

UMR 8590 CNRS/PARIS 1/ENS

Thèse pour l'obtention du grade de
docteur de l'Université Paris 1
Panthéon-Sorbonne

Présentée et soutenue publiquement par :
Hector ZENIL

L'APPROCHE ALGORITHMIQUE DE L'ALÉATOIRE

Peut-elle expliquer la nature organisée du monde ?

Directeur de thèse : M. Jean MOSCONI
Professeur émérite Université de Paris 1

JURY :

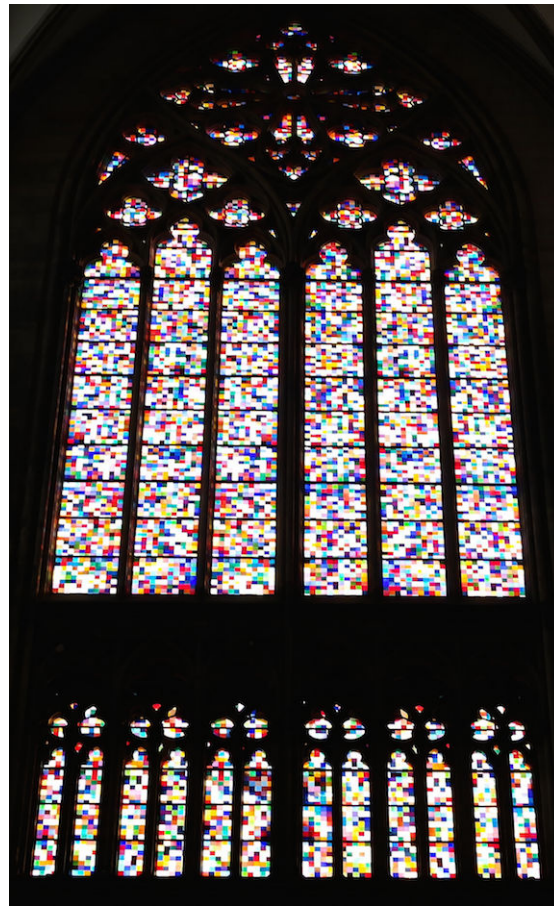
Andrew ADAMATZKY	U. of the West of England	<i>Examineur</i>
Barry COOPER	University of Leeds	<i>Rapporteur</i>
Jean-Paul DELAHAYE	Université de Lille 1	<i>Examineur</i>
Cyrille IMBERT	Université de Lorraine	<i>Examineur</i>
Jean MOSCONI	Université de Paris 1	<i>Directeur</i>

Date de soutenance : 14 Novembre 2015

Code : 2015-11 ZENIL Lap

INSTITUT D'HISTOIRE ET DE PHILOSOPHIE DES SCIENCES ET DES TECH-
NIQUES (IHPST)
UMR 8590
13, RUE DU FOUR, 75006 PARIS

UNIVERSITÉ DE PARIS 1 – PANTHÉON-SORBONNE
ÉCOLE NORMALE SUPÉRIEURE ULM
CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE (CNRS)



“...Richter used a specially developed computer programme to determine the random arrangement of the coloured squares in one half of the window. The resulting arrangement in these lancet windows was then mirrored... As a result, this section of the window seems even more magical than almost any other window in the cathedral...” Barbara Schock-Werner

“...The vibrant kaleidoscope of colour does not force any specific interpretations on the beholder ; it is open and allows light to shine through the myriad nuances of colour... The process of artistic creation dares to bring randomness and rationale together ; to combine the unpredictable with the mind that can select, arrange, and determine. The fruits of this interaction are beauty, an incomprehensible order, and a creative origin that the artist cannot ultimately explain, even though he or she is responsible for making the all-important creative decision... The beholder cannot truly grasp the coloured squares ; patterns seem to appear only to disappear again, revealing new patterns.

Ultimately it is the mysterious and the inexplicable that win through. The light sparkles, shines, and is resplendent in this sea of colour.” Josef Sauerborn

Excerpts (2008) from Barbara Schock-Werner [Head architect of the cathedral] and Josef Sauerborn [Chaplain to the arts community in the Archdiocese of Cologne] about Gerhard Richter’s stained glass window commissioned for the south transept of Cologne’s cathedral in Germany in 2003.

Picture by H. Zenil, Cologne, Germany ; 2015.

Remerciements

Après avoir fini un doctorat en informatique fondamentale à Lille et d'avoir entrepris trois postes de recherche postdoctoral (Sheffield, Karolinska et Oxford) je tiens à remercier les nombreuses personnes qui m'ont donné les conseils utiles, au long de mes études et de ma recherche, pour construire ma pensée d'aujourd'hui. Je tiens tout d'abord à remercier mon directeur de thèse M. Jean Mosconi pour avoir accepté d'encadrer mon travail de recherche. À mes parents pour leur appui inépuisable et aux membres du jury de thèse Andrew Adamatzky, Barry Cooper, Jean-Paul Delahaye et Cyrille Imbert. Je voudrais aussi remercier Bernard François pour son aide à la rédaction et pour ses commentaires précis, Antonio Rueda-Toicen pour son aide à la mise en page et Adrian German pour sa collaboration dans l'organisation de l'un des groupes de discussion, la réédition du travail de Konrad Zuse et la transcription (en anglais) des textes des groupes de discussion que j'ai organisé pour cette thèse. Je tiens aussi à remercier Horst Zuse, le fils de Konrad Zuse, pour son autorisation à republier la nouvelle édition du travail de son père, ainsi que le MIT et Springer pour la même raison, et aussi à Roger Penrose pour avoir écrit le préface de mon recueil *A Computable Universe* publié par World Scientific Publishing Company et l'Imperial College Press. Finalement, je remercie mes collaborateurs du groupe de travail de *La Nature Algorithmique* (Algorithmic Nature Group/Lab <http://algorithmicnaturelab.org/>) du Laboratoire de Recherche Scientifique LABORES Pour les Sciences Naturelles et Numériques, Jean-Paul Delahaye (qui a été aussi mon directeur de thèse d'informatique), Joost Joosten, Fernando Soler Toscano, Nicolas Gauvrit, Jürgen Riedel, parmi d'autres. Et, pour de raisons diverses, aussi à Jesper Tegnér, Peter Minary, James Marshall, Stephen Wolfram, Gregory Chaitin, Matthew Szudzik, Frederico Meinberg, Jason Cawley, Todd Rowland, Gordana Dodig-Crnkovic, Vela Velupillai, Selmer Bringsford, Jeremy Avigad, Wilfried Sieg, Kevin Kelly, Cristian Calude, Roger Penrose, Catherine Boucher, Maurice Margenstern, Barry Cooper, Andrew Adamatzky et Mariana González Castellón Gutiérrez, ainsi qu'au Consejo Nacional de Ciencia y Tecnología (CONACyT), l'IHPST, le Foundational Questions Institute, la Silicon Valley Community Foundation et la John Templeton Foundation, parmi d'autres.

Hector Zenil

Resumé

L'APPROCHE ALGORITHMIQUE DE L'ALÉATOIRE, PEUT-ELLE EXPLIQUER LA NATURE ORGANISÉE DU MONDE ?

Fondements de la complexité algorithmique

On entreprend une étude approfondie des fondements de la théorie de la complexité algorithmique qui donne sens au concept de l'aléatoire fini (partie I). On analyse les axiomes sur lesquels la complexité et le concept d'aléatoire sont fondés ainsi que les théorèmes fondateurs de la théorie, notamment le théorème d'invariance et le théorème de codage algorithmique de la théorie de la probabilité algorithmique. Je passe en revue certains sujets pertinents quant au débat philosophique sur l'information, en tenant compte des développements de la théorie algorithmique de l'information. Par exemple, je vais proposer l'hypothèse que la signification possède une profondeur, au sens de la profondeur logique de Bennett, et que avec la probabilité algorithmique de Solomonoff et Levin on peut fournir la stabilité nécessaire pour une définition algorithmique robuste de la signification, une définition prenant en considération l'interprétation et la connaissance propre du destinataire encodées dans l'histoire attachée au message.

Le monde est-il aléatoire ?

En s'appuyant sur la partie I précédente et les débats organisées dans le contexte de cette thèse (appendice), on examinera les théories informationnelles et digitales du monde liées au monde physique, du "pancomputationalisme" aux hypothèses de Zuse, Feynman, Wheeler, Fredkin, Wolfram, Penrose et Lloyd. On utilisera des méthodes algorithmiques qui s'appliquent aux théories de l'information pour essayer de tirer des conclusions qui puissent nous fournir des explications vraisemblables de certains phénomènes (notamment ceux de l'émergence d'ordre). Si on veut savoir si le monde est algorithmique par nature, il faudrait d'abord spécifier ce à quoi pourrait ressembler

un monde algorithmique. Que l'univers soit algorithmique n'implique pas nécessairement que le monde soit banal, prévisible ou déterministe (partie II).

Une conception algorithmique de l'univers

En effet, si le monde est à tous points de vue un monde algorithmique, les structures en lui doivent se ressembler, avec des distributions de motifs se ressemblant aussi. Pour démontrer cela, Nous avons conçu et réalisé une série d'expériences afin de produire des données par des moyens purement algorithmiques, pour les comparer avec des ensembles de données produites par plusieurs sources physiques. Je vais discuter ces expériences et ce qu'elles signifient mais je vais renvoyer le lecteur aux journaux où j'ai publié ces travaux plutôt techniques qui ont été des parties de ma thèse de doctorat en informatique fondamentale à Lille.

Je vais utiliser la théorie algorithmique de l'aléatoire pour argumenter en faveur de l'hypothèse d'un monde algorithmique. Je vais proposer l'idée que l'univers est algorithmique par nature, mais en parlant de ce dont il est fait, mais en parlant de la manière dont il se déploie. Au cœur de l'argument se trouve les concepts de complexité et probabilité algorithmique, qui servent à comparer la façon dont les structures sont distribuées dans le monde réel avec la façon dont elles le sont dans un monde digital simulé. La vision algorithmique révèle la beauté intrinsèque de la convergence des modèles digitaux. Et que cette approche peut même aider à expliquer de phénomènes cognitives.

Je propose deux mesures pour tester l'hypothèse algorithmique, celle de l'*algorithmicité* des distribution des structures dans le monde réel ou comment elles semblent être distribuées lorsqu'on les compare avec une distribution purement algorithmique (la *distribution universelle*,) et une mesure de *programmabilité* (partie III) applicable à des systèmes individuels pour déterminer le degré de ressemblance au comportement d'une machine universelle (de Turing).

Summary

THE ALGORITHMIC APPROACH TO RANDOMNESS, CAN IT EXPLAIN THE ORGANISED STRUCTURE OF THE WORLD?

Foundations of the theory of algorithmic complexity

Part I includes an introduction where I explore some of the foundations of the theory of algorithmic information that underlie the concept of algorithmic randomness. I also present an analysis of the results and theorems upon which the concepts of finite algorithmic complexity and finite randomness are founded, notably the theorem of invariance and the theorem of algorithmic coding from the theory of algorithmic probability. I survey some aspects relevant to the philosophical discussion of the concept of information, centrally taking into account developments in algorithmic information theory. For example, I will propose that Bennett's Logical Depth and Solomonoff-Levin's algorithmic probability can contribute toward the foundation of an algorithmic definition of meaning, one which takes into consideration the interpretation and knowledge of the receiver that is encoded in the history attached to the message.

Is the world random?

Using as a basis Part I and the round tables organised around this thesis (Appendix), we will examine the leading theories of an informational and computational world, from the so-called "pancomputationalism", to considering the hypotheses of Zuse, Feynman, Wheeler, Fredkin, Wolfram, Penrose and Lloyd. We will use algorithmic methods applied to the theories of information to try to outline possible explanations of observable phenomena, notably the emergence of order and structure. But if we want to know whether the world is algorithmic in essence, we should define what we mean by an algorithmic world, and attempt to imagine what it would look like. We will see that the algorithmic world view reveals an inherently beautiful

convergence of digital models. However, I propose that an algorithmic universe does not necessarily imply a trivial, predictable or deterministic one (Part II).

An algorithmic view of the world

Indeed, if the world is to all intents and purposes algorithmic, its structures should resemble those of a simulated algorithmic world, showing similar distributions of patterns. To test this, I conceived and carried out a series of experiments in order to produce data from purely algorithmic processes to compare with datasets from different real-world sources. I will discuss the experiments and what they mean, while also referring the reader to the original journal articles where I published these ideas, some of which were part of my previous PhD thesis in Theoretical Computer Science at Lille.

I will then use the algorithmic theory of randomness to defend the hypothesis of an algorithmic world (Part III). I propose the idea that the universe is algorithmic in essence, not by virtue of the matter it is composed of (which raises questions of continuity and size), but by virtue of the way in which this matter is deployed. At the heart of the argument is the concept of algorithmic probability, which serves to compare the way in which patterns are distributed in real-world and in simulated digital datasets, and even explain animal and human cognitive phenomena.

I propose two testable measures for my algorithmic view. The first is *algorithmicity*, or how patterns in the physical world seem to be distributed when compared to the so-called *universal distribution*, a distribution from algorithmic probability, while the second is a measure of *programmability* that I will claim is also applicable to individual systems (in Part III) to determine the degree to which a system can behave as a universal (Turing) machine.

Resumen

¿PUEDE EL ENFOQUE ALGORÍTMICO DE LA ALEATORIEDAD EXPLICAR LA ESTRUCTURA ORGANIZADA DEL MUNDO?

Fundamentos de la teoría de la complejidad algorítmica

La Parte I consta de una introducción en donde se exploran algunos de los fundamentos de la teoría de la información algorítmica, base del concepto de la aleatoriedad algorítmica. Asimismo se presenta un análisis de los resultados y teoremas sobre los que se fundamentan los conceptos de la complejidad algorítmica finita y la aleatoriedad finita, en particular el teorema de la invariabilidad y el teorema de la codificación algorítmica desde la teoría de la probabilidad algorítmica. Se estudian algunos aspectos importantes para el debate filosófico del concepto de la información; principalmente, se toman en cuenta los avances de la teoría de la información algorítmica. Por ejemplo, se propone que la profundidad lógica de Bennett y la probabilidad algorítmica de Solomonoff-Levin pueden contribuir a la creación de una definición algorítmica de significado, que tenga en cuenta la interpretación y el conocimiento del receptor que se encuentra codificado en el historial adjunto al mensaje.

¿Es aleatorio el universo?

Con base en la Parte I y las mesas redondas que se organizaron para esta tesis (Apéndice), se analizan las principales teorías sobre un mundo informático y computacional, pasando desde el llamado “pancomputacionalismo” hasta las hipótesis de Zuse, Feynman, Wheeler, Fredkin, Wolfram, Penrose y Lloyd. Se usan los métodos algorítmicos aplicados a las teorías de la información, para tratar de delinear posibles explicaciones de fenómenos observables, en particular el surgimiento del orden y de la estructura. No obstante, si se busca saber si el mundo es algorítmico en esencia, se debe definir qué significa el concepto de un mundo algorítmico e intentar imaginar cómo luciría; la

perspectiva algorítmica del mundo revela una convergencia intrínsecamente hermosa de modelos digitales. Sin embargo, lo que se propone en el presente trabajo es que un universo algorítmico no necesariamente tiene que ser trivial, predecible o determinista.

Una perspectiva algorítmica del universo

Si para efectos prácticos el mundo es algorítmico, sus estructuras deberían parecerse a las de un mundo algorítmico simulado, con una distribución similar de patrones. Para probar esto, se propone y diseña una serie de experimentos para producir datos a partir de procesos puramente algorítmicos, para hacer una comparación con conjuntos de datos de diferentes fuentes del mundo real; se habla acerca de los experimentos y lo que significan, mientras que también se hace referencia a los artículos originales en donde el autor de esta tesis publicó estas ideas, algunas de las cuales forman parte de su tesis anterior de doctorado en ciencia computacional teórica en Lille.

Después, se usa la teoría algorítmica de la aleatoriedad para sostener la hipótesis de un mundo algorítmico (Parte III). Se propone la idea de que el universo es en esencia algorítmico, no en virtud de la materia de la que está compuesto (lo que genera preguntas sobre continuidad y tamaño), sino debido a la forma en que esta materia se encuentra distribuida. El tema central de este argumento es el concepto de la probabilidad algorítmica, que sirve para comparar la forma en que los patrones se distribuyen en el mundo real y en conjuntos de datos digitales simulados, e incluso para explicar los fenómenos cognitivos humanos y animales.

Se establecen dos medidas concretas para investigar la perspectiva algorítmica que se propone. La primera es la de *algoritmicidad*, es decir, cómo los patrones del mundo físico parecen distribuirse cuando se les compara con la llamada distribución universal (una distribución de la probabilidad algorítmica), mientras que la segunda es una medida de *programabilidad*, que en el presente documento se presenta como aplicable a los sistemas individuales (Parte III) para determinar el grado hasta el que un sistema puede comportarse como una máquina (de Turing) universal.

Several chapters and sections of this thesis have been published in journals as part or whole, a few others have received a prize or a special mention. The following table indicates what chapter roughly corresponds to what published work(s). For the same reason, some redundancy is expected as they were originally intended to be self-contained pieces and they may not necessarily map one-to-one :

Publication	Journal/Awarding institution	Chapter
Textbook : Philosophy of Information : An Introduction	<i>Society for the Philosophy of Information</i> in the UK, 2013	1
Empirical Encounters with Computational Irreducibility and Unpredictability	<i>Minds and Machines</i> , 2012	2
What is Nature-like Computation ? A Behavioural Approach and a Notion of Programmability	<i>Philosophy & Technology</i> , 2013	4
A Turing Test-Inspired Approach to Natural Computation	Proceedings of <i>Turing in Context II</i> Historical and Contemporary Research in Logic, 2012	4
The World is Algorithmic or Mostly Random	<i>3rd prize FQXi Contest</i> , 2011	5
An Algorithmic Approach to Information and Meaning	<i>APA Newsletter on Philosophy and Computers</i> , 2011	8
The Information-theoretic and Algorithmic Approach to Human, Animal and Artificial Cognition	Representation and reality : Humans, animals and machines (in press)	9
* Randomness Through Computation	<i>World Scientific/Imperial College Press</i> 2011	10, 11
* A Computable Universe	<i>World Scientific/Imperial College Press</i> , 2013	10

Marked with * are books for which I have served as editor. They have been declared bestsellers by World Scientific/Imperial College Press. The book *A Computable Universe* was published with the support and in commemoration of the Alan Turing Year in 2012.

Table des matières

1	Introduction	xxv
I	La calculabilité pour expliquer l'aléatoire	1
1.1	De la théorie classique des probabilités à la théorie du calcul . . .	3
1.1.1	La probabilité classique d'une suite	3
1.1.2	La machine de Turing	5
1.1.3	Les notions intuitives de l'aléatoire	5
1.1.4	Le contenu d'information d'un objet individuel	6
1.2	La complexité algorithmique	10
1.2.1	Complexité de Kolmogorov	11
1.3	L'aléatoire algorithmique par incompressibilité	14
1.3.1	Les martingales effectives ou l'aléatoire selon Schnorr . .	16
1.3.2	Les tests statistiques effectifs ou l'aléatoire selon Martin-Löf	18

1.4	Convergence des définitions	19
1.5	La nature subjective de l'aléatoire pour des suites finies	21
1.6	La valeur ontologique et épistémologique des définitions	22
1.6.1	Le problème de l'aléatoire local	23
1.6.2	Ontologie et épistémologie des définitions de l'aléatoire	23
1.6.3	La valeur ontologique et épistémologique des définitions	24
1.7	La définition de la complexité algorithmique est-elle robuste ? .	25
1.7.1	Le choix de la machine universelle est important	27
1.8	Resumé des propriétés de la complexité algorithmique	28
1.9	Une variante commode de K	31
1.10	Resumé des propriétés de C et K :	33
1.11	La probabilité algorithmique et la distribution universelle de Levin	33
1.11.1	La complexité "préfixe" pour une mesure de probabilité algorithmique	33
1.11.2	Prévisibilité optimale, probabilité de Solomonoff et la demi-mesure de Levin	34
1.11.3	De la métaphore du singe dactylographe au singe pro- grammateur	37
1.11.4	Complexité et fréquence	38
1.12	Le scepticisme sur la nature (in)déterministe du processus gé- nérateur	40

1.12.1	Vers une définition stable ou comment tester la nature algorithmique du monde	42
1.12.2	Convergence en distribution	42
1.12.3	Le nombre à la sagesse infinie : le nombre Omega (Ω) de Chaitin	44
1.12.4	Principe heuristique de Chaitin	47
1.13	La profondeur logique de Bennett	49
1.14	La contingence de l'aléatoire quantique	50
1.15	La structure cachée des nombres	51
1.15.1	Un îlot dans l'océan de l'irréductibilité Raccourcis de π et l'irréductibilité de Wolfram	52
1.15.2	Principes d'irréductibilité	53
1.15.3	L'aléatoire intrinsèque	57
1.16	Resumé de la première partie	58

II De l'incalculable à l'irréductible 61

2 Rencontres Empiriques avec l'Irréductibilité et l'Imprévisi- bilité dans le Calcul 63

2.1	Une approche informatique expérimentale d'un débat épisté- mologique	63
2.2	Prévoir le comportement de systèmes de calcul déterministes .	65

2.3	Mesure de l'irréductibilité	68
2.4	L'expérience	70
2.5	Formalisme	71
2.6	Définition d'une fonction calculée et choix de la durée avant coupure	72
2.7	Le programme prédicteur	74
2.8	Résultats	76
2.8.1	Ensembles décidables	76
2.9	Classification des séries	77
2.10	Processus d'achèvement	80
2.10.1	Sélection des séries à achever (ou à compléter)	80
2.11	Comparaison entre prédictions et résultats réels	81
2.12	Une seconde tentative de prédiction	82
2.13	Expérimentations plus larges avec (4,2)	82
2.14	Discussion sur l'encodage des résultats	83
III	Vers une théorie algorithmique du monde	87
3	Des univers computables	89
3.1	Qu'est-ce que la computation ? Comment calcule la nature ? .	90
3.2	L'approche algorithmique	92

3.3	La complexité des systèmes vivants	94
3.4	Comment s'intègre le Quantique	98
3.5	Vers une théorie informatique de la physique	103
3.6	Une approche algorithmique du problème de l'ajustement pré- cis (fine-tuning)	105
3.7	Les trous noirs en tant que possibles compresseurs avec pertes de données	106
3.8	Cartographie simplifiée des visions (in)calculables du monde .	108
4	Vers une vision algorithmique du monde	113
4.1	La structure née de l'aléatoire	116
4.2	Complexité algorithmique dans le monde réel	118
4.3	La place de la mécanique classique	119
4.4	Les lois physiques comme des programmes informatiques . . .	121
4.5	Pas de séparation entre programmes et données	122
4.6	La signification d'algorithmique	123
4.7	Le modèle du calculateur quantique	124
4.7.1	L'arc-en-ciel des bruits colorés en tant que pseudo- aléatoires.	126
4.7.2	Divergence des définitions analogiques	128
4.7.3	Caractéristiques concevables pour les modèles de l'uni- vers	129

4.7.4	Hypothèse de l'aléatoire contre hypothèse de l'ordonné	131
4.8	Une interprétation algorithmique de la mécanique quantique	134
4.9	La place de l'information dans le réel	136
4.10	La nature algorithmique du monde	137
4.11	Comment la mécanique quantique implique le déterminisme	141
4.11.1	La question de la liberté de choix	144
4.11.2	Interprétation informationnelle de la mécanique quantique	145
4.11.3	Information quantique et fondements informationnels de la physique moderne	146
5	Calcul : Production d'ordre à partir de l'aléatoire	151
5.1	Information et énergie	152
5.1.1	Computation réversible	153
5.2	La question de l'exécution	153
5.3	Vers un univers constitué de chaînes de bits	155
5.4	Un monde digital simulé	156
5.5	L'ADN est simple, mais le déploiement des protéines est complexe	159
5.6	La vie est robuste car elle suit des règles	161
5.7	Les données empiriques sont compressibles	164
5.8	Révéler la machinerie	167

5.9	Vers une conception informatique du monde	170
6	Mécanique algorithmique des processus du vivant	173
6.1	Émergence dans les systèmes biologiques auto-assemblés . . .	174
6.2	L'émergence comme un phénomène algorithmique	178
6.3	Probabilité algorithmique et robustesse biologique	180
6.4	Modèles computationnels de taux d'évolution biologique . . .	183
6.4.1	Le coût du traitement de l'information et de l'aptitude	184
6.5	Recherche et formation de structures biologiques	187
6.6	L'universalité de Turing et le rôle de la complexité décrite dans les systèmes vivants	190
6.6.1	Temps d'arrêt, et émergence épistémique contre émer- gence ontologique	191
6.7	Rôle et nature de l'aléatoire dans les processus biologiques . .	192
6.7.1	Compétition algorithmique et prévision	194
6.8	L'évolution biologique comme un processus algorithmique . . .	195
6.9	Notes du chapitre	198
7	Une approche algorithmique de l'information et de la signi- fication	201
7.1	Qu'est-ce que l'information	201
7.2	Communication, diversité et complexité	204

7.3	Données + un programme est un message + interprétation . .	205
7.4	Contenu informatif et signification	207
7.5	Contenu informatif et signification intrinsèque	208
7.6	La signification est logiquement profonde	210
7.7	Une définition algorithmiquement robuste de la signification .	211
7.8	Un aléatoire fini (tout comme la signification) se situe dans l'œil de l'observateur	212
7.9	Vers un agenda philosophique	213
7.10	Bases sur lesquelles s'entendre	216
7.11	Remarques de chapitre	217
7.11.1	Remerciements	218
8	Une approche comportementaliste du calcul naturel et un test de programmabilité	219
8.1	Introduction	219
8.2	Un fondement classique pour la computation non- conventionnelle	221
8.2.1	Une approche comportementale de la computation . . .	221
8.2.2	La computation digitale en tant que computation na- turelle	223
8.3	Une approche inspirée du test de Turing, pour la computation	225

8.3.1	La complexité algorithmique comme mesure approxi- mative de la programmabilité	229
8.3.2	Une approche comportementale de la computation . . .	232
8.4	Une taxonomie de la computation	234
8.5	Examen des objections possibles	241
8.5.1	Objections techniques	242
8.5.2	Objections sur les fondements	245
8.6	Remarques en conclusion	257
9	L'approche algorithmique de la cognition	259
9.1	Le modèle algorithmique de l'esprit	259
9.1.1	Le test de Turing est trivial, mais l'esprit est algorithmique	262
9.2	La complexité algorithmique comme modèle de l'esprit	267
9.2.1	Des tendances à Bayes	268
9.2.2	La méthode du théorème du Codage	270
9.2.3	La Méthode de Décomposition par Blocs	273
9.3	Cognition et Complexité	274
9.3.1	Mémoire de travail	275
9.3.2	Perception de l'aléatoire	277
9.3.3	Culture et structure	281
9.4	Remarques de conclusion	283

10 Conclusion générale	285
 IV Appendice	 291
 A Groupe de discussion : L'univers est-il aléatoire ?	 293
i Complexité versus aléatoire	294
ii Pouvons nous contrôler l'univers ?	297
iii Esprits, observateurs et l'univers	300
iv Esprits qui communiquent	303
v Le problème de l'Intelligence Artificielle	308
vi Les mathématiques sont-elles inévitables ?	310
vii L'univers est-il calculable ?	314
viii La chasse à l'univers qui est le nôtre	320
ix Le futur des métamathématiques	324
 B Groupe de discussion : Qu'est-ce que le Calcul ? Est-ce que la Nature calcule (et comment) ?	 335
i Computation quantique vs computation classique	336
ii L'idée de computation	338
iii L'univers est-il un calculateur ?	349
iv L'univers est-il discret ou continu ?	358
v L'univers est-il aléatoires ?	363

vi	information vs matter	372
vii	Questions du public, sans modérateurs	378
C Open Discussion on A Computable Universe		391
D Calculating Space (<i>Rechnender Raum</i>)		453
i	Introduction	455
ii	Introductory observations	457
ii.1	Concerning the Theory of Automata	457
ii.2	About Computers	459
ii.3	Differential Equations from the Point of View of the Automaton Theory	462
ii.4	Maxwell Equations	465
ii.5	An Idea about Gravitation	467
ii.6	Differential Equations and Difference Equations, Digi- talization	467
ii.7	Automaton Theory Observations of Physical Theories .	469
iii	Examples of digital treatment of fields and particles	473
iii.1	The Expression “Digital Particle”	473
iii.2	Two-Dimensional Systems	481
iii.3	Digital Particles in Two-Dimensional Space	484
iii.4	Concerning Three-Dimensional Systems	489

iv	General considerations	491
iv.1	Cellular Automata	491
iv.2	Digital Particles and Cellular Automata	493
iv.3	On the Theory of Relativity	493
iv.4	Considerations of Information Theory	495
iv.5	About Determination and Causality	502
iv.6	On Probability	506
iv.7	Representation of Intensity	507
v	Conclusions	510
E	Afterword to Konrad Zuse's Calculating Space	513

Chapter 1

Introduction

I have developed the ideas introduced and discussed in this thesis over several years. In the process I finished a PhD in Theoretical Computer Science at the University of Lille 1, Nord-Pas-de-Calais, France under supervision of Jean-Paul Delahaye, and I have undertaken postdoctoral positions at the Behavioural and Evolutionary Theory Lab of the Department of Computer Science of the University of Sheffield in the United Kingdom and at the Unit of Computational Medicine at the Karolinska Institute in Stockholm, Sweden. My ideas in this time, even if they have evolved, have remained in the same direction. I believe that the world is algorithmic in a fundamental fashion. Moreover, I believe this is not a controversial point of view and its only possible controversial side is the way in which it is made explicit. However, all science works under this assumption and technology has been profiting from its results for a long time. Science studies regularities that are then described in a model of the world to make predictions. The assumption of science that the world behaves in some regular, well-behaved fashion, is unquestionable, and indeed, according to classical mechanics if we are not able to predict any future event with full accuracy is only because of our measurement limitations to represent the exact current state of the world. Only certain interpretations of quantum mechanics may suggest truly non-deterministic randomness but the ultimate answer is unresolved.

That the world is algorithmic has its own story starting from explicit computational and even digital worldviews but the worldview introduced and exploited in this thesis is novel and had not been studied and followed before. Some other worldviews may imply this or other algorithmic view, such as digital physics, but what I claim here and do defend is that the algorithmic view I introduce has testable properties and an important predictive power. While the term *algorithmic* is opposed to that of *random*, the mathematical

field that studies randomness is, paradoxically, called *algorithmic randomness* and I claim it is the right framework to ask about the essence of nature and the organised nature of the universe. Potentially misleading names are common currency in mathematics. Another relevant example is the main theory on which algorithmic randomness is based upon and that is called *computability theory* which turns out to study *uncomputable* objects (e.g. infinite strings/sequences).

One similar view to mine, although not the popular trivialized version that is commonly attributed to him (that the world is a cellular automaton), is that of Stephen Wolfram, who suggests the universe may be the product of a simple (I take it as short) computer program, and has specifically proposed a candidate algorithm based on evolving causal networks for such a program (even if the program is Turing computable it does not mean that it is carried out by a Turing machine, nor a cellular automaton—CA—as he explains that the synchronous time in a CA would make it difficult to lead to phenomena such as relative time needed in general relativity). Indeed, a digital computational world would imply the algorithmic world I am here presenting, but my algorithmic view also makes it more precise, as I will argue it can be even tested, and I have tested it myself with some interesting results. But more important, science tests the algorithmic view everyday with palpable success. To illustrate that the question can be experimentally approached, I have proposed to look at the distribution of patterns in the world as compared to a distribution of patterns that an algorithmic world would produce, according to for example the theory of algorithmic probability. It would amount to statistical evidence if nature distributes these patterns similarly to how computer programs would do. On the other hand, I advocate for what I call the *programmable* properties of the world, epitomized by a measure of *programmability* that I introduce and that will help us to classify objects and delete frontiers between what may seem completely different systems in the world.

In this thesis I explore these questions, and I do so by looking at the mathematical theory that can deal with the information content and randomness of finite objects. The field of algorithmic randomness and Kolmogorov complexity can be divided into those researchers that study infinite sequences and their complexity in connection to computability and those that study finite strings and are closer to actual applications in various fields of science. Infinite sequences do not occur in nature as far as we can tell (or measure), and a better approximation to possible algorithmic properties of the world is, from my perspective, by studying the properties of the finite objects it contains. Nevertheless, the frontiers of what can be computable

and what is not, and between random and non-random objects have also shed important light into fundamental questions. Some of these algorithmic properties of computation that can tell us about the type of randomness in the world can be studied by means of what is called *algorithmic probability*, a subarea of the fields of algorithmic randomness and Kolmogorov complexity. While I will be focusing on finite strings, other dissertations have recently been defended covering the foundations of the infinite side of the theory of algorithmic randomness [Porter, 2012] and the philosophical implications of algorithmic probability in the problem of inference [Sterkenburg, 2013] or induction [Vallinder, 2012] (as originally denominated by R. Solomonoff).

The opposing theory of my thesis, that is the view that the world is rather random, more random or mostly random than algorithmic, is from my perspective the theory (in the sense of model hypothesis of the world) from classical information theory epitomized colloquially by what is known as *Boltzmann brains*, and that has been reformulated in the context of algorithmic complexity by Cristian Calude, one of my thesis advisors of my PhD thesis in Computer Science. I have included a translation of a paper that Cristian Calude wrote for one of my edited volumes [Calude et al., 2012], precisely in opposition to my algorithmic view. Of course a large part of this thesis is devoted to explaining what it is meant by *algorithmic*. J. W. McAllister, for example, advocates [McAllister, 2003] for a random world, not one underlying ours but one in which we currently live in, claiming that scientific models only capture a small part of the regularities, leaving on the side most of what we consider noise and that in his view is the random signature of the universe and its most important feature. I differ with both accounts, that of Calude and McAllister. In this thesis, I present my arguments by exploring some of the most striking consequences of the finite study of algorithmic randomness.

An even more radical view is that of Roger Penrose. I asked him to write a preface for a book I edited for World Scientific under the title *A Computable Universe*, with authors contributing to world views related or removed from my algorithmic view. Among the worldviews that Penrose compares is my own, as defended in this thesis. I have also included in the appendix of this thesis the translation of Konrad Zuse's *Calculating Space* in a new edition rewritten by me and Adrian German in $\text{\LaTeX}$ with some corrections and an all modernized notation as a contribution of, hopefully, historical value. The renewed version was produced with permission from Zuse's family, of Springer Verlag, and of the MIT who did the original English translation from an outdated journal of the Verlag house.

Finally, also in the Appendix, I include the transcription and translation

to French of two discussion tables I organised in the U.S. around my thesis subject on the nature of nature, either random or computable, featuring some of the most prominent proponents of various competing worldviews, including Stephen Wolfram himself; Gregory Chaitin, the only living founding father of algorithmic complexity (who also served in my evaluation committee of my PhD thesis in computer science); Cris Calude (my co-supervisor of my PhD in computer science); Ed Fredkin, the pioneer of digital physics; Tony Leggett (the Nobel prize in physics); Tom Toffoli, another pioneer in digital physics and reversibility; Karl Svozil, a leader in the foundations of quantum mechanics, and Rob de Ruyter, an expert in biology. The conference organised by me and Prof. A. German included Seth Lloyd, Charles Bennett, Lov Grover and David Deutsch (slides, video and sound are all available online at <http://www.cs.indiana.edu/~dgerman/2008midwestNKSconference/whatsnew.html>). The last chapter of my thesis is a written discussion between me and most of these authors, in a sort of follow up and a reply of contrasting views. Only my answers and their questions were left untranslated from their original English language but the roundtable discussions and some other contributions were translated to French and their original English documents are also available online.

Première partie

La calculabilité pour expliquer l'aléatoire

Brève introduction à la théorie du calcul et à la théorie algorithmique de l'information

Rien n'arrive dans le monde qui soit absolument irregulier. Pour ce qui est de la simplicité des voyes de Dieu, elle a lieu proprement à l'égard des moyens, comme, au contraire, la varieté, richesse ou abondance y a lieu à l'égard des fins ou effets.

Leibniz, Discours de métaphysique, VI, 1686.
évoqué par Gregory Chaitin.

1.1 De la théorie classique des probabilités à la théorie du calcul

1.1.1 La probabilité classique d'une suite

Selon la théorie classique des probabilités, sous une distribution uniforme, toutes les suites à n chiffres (par exemple pour des suites qui encodent les résultats d'un jeu de pile ou face avec une pièce non truquée) :

[illegible]

2 :PFPFPFPFPFPFPFPFPFPFPF

$\mathcal{J} : PFFPPFPPPPFFFFPPFPPFPFPFFFFP$

1. P=pile, F=face

ont la même probabilité $P(s) = 1/2^n$ comme résultat de lancer la pièce n fois. Alors qu'évidemment si on obtient la première on supposera que la pièce était truquée, et si on tombe sur la deuxième on la soupçonnera aussi, tandis que si on tombe sur la troisième, cela semblera plus normal. On voudrait formaliser l'idée qu'on n'attend pas de régularités d'un processus dans une suite d'événements résultant du hasard. Le fait de douter de la nature aléatoire de certaines suites, n'est qu'un préjugé selon la théorie classique des probabilités. si on peut bien montrer facilement que la probabilité qu'une suite de longueur 26 tirée à pile ou face ne contienne aucun F (comme la suite 1) est extrêmement faible et que la probabilité qu'une suite de longueur n ne contienne ni PP ni FF (comme la suite 2), il faut considérer des tests sur tout l'ensemble des suites pour que la probabilité devienne vite très faible quand n croît et qu'on puisse distinguer de tels cas avec des tests triviaux. Par contre, même avec des raffinements dans cette direction, par exemple la notion de suite normale de Borel, on n'arrive pas à donner une caractérisation satisfaisante de l'aléatoire comme on verra plus tard.

Malgré tout, notre intuition persiste à dire qu'une suite telle que 000000... ou 111111... (pour, par exemple, 0 = pile, 1 = face) n'est pas aléatoire. La théorie algorithmique de l'information donne une explication à cette intuition par le biais du concept de contenu d'information. On voudrait formaliser l'idée qu'on n'attend pas de régularités d'un processus dans une suite d'événements résultant du hasard, et que chaque suite est capable de contenir de l'information qui la différencie.

Les définitions que la théorie algorithmique de l'information fournit sont des définitions de type négatif, c'est-à-dire qu'on donne une définition en termes de calculabilité de l'objet non aléatoire, puis on définit un objet aléatoire comme étant un objet qui ne possède pas cette propriété même si on ne le montre pas.

La théorie de l'aléatoire algorithmique se base sur le modèle de calcul de Turing. La thèse de Church-Turing, largement acceptée comme vraisemblable, affirme qu'est "calculable" ce qui est "calculable par une machine de Turing" (ou tout autre modèle de calcul équivalent). On dit alors que les fonctions qu'on peut calculer avec une machine de Turing ou avec un algorithme sont les fonctions calculables (on dit aussi récursives). Les fonctions usuelles ($n \rightarrow n^2$, $n \rightarrow n - i^{\text{ème}}$ nombre premier, etc.) sont calculables.

1.1.2 La machine de Turing

Pour comprendre les fondements de l'AIT il faut comprendre le modèle de machine de Turing. Ce modèle [Turing, 1936] est un modèle abstrait d'ordinateur composé d'un mécanisme de calcul et d'un ruban sur lequel la machine écrit et qu'elle peut effacer en déplaçant une tête de lecture-écriture. Chaque machine de Turing est associée à un programme qui détermine complètement, dans tous leurs détails, les opérations qu'elle effectue. Alan Turing démontre qu'il existe un type de machine de Turing (dite universelle) capable de simuler toute autre machine de Turing.

Le problème de l'arrêt

On démontre qu'il existe des fonctions non calculables : la première d'entre elles est la fonction qui, pour tout programme donné P, indique (a) si le programme finit par s'arrêter ou (b) s'il continue indéfiniment à calculer. Effectivement, Turing démontre que cette fonction n'est pas calculable, c'est ce qu'on nomme l'indécidabilité du problème de l'arrêt. En d'autres termes, il n'existe aucune machine de Turing qui sait déterminer si une autre machine va s'arrêter pour un programme avec une entrée quelconque. Beaucoup d'autres problèmes sur les machines de Turing ne sont pas décidables.

La pertinence du langage binaire

Le langage binaire permet d'encoder des événements comme ceux du pile ou face mais aussi toute autre série d'événements discrets par le biais d'un changement de base. Par exemple, si on étudie les sorties d'un dé qui peut donner six résultats possibles, on peut les encoder chacun par un symbole différent et les transformer en binaire.

1.1.3 Les notions intuitives de l'aléatoire

Dans la littérature du hasard, on associe l'aléatoire de manière intuitive avec les notions de :

- se produire sans loi.
- ne pas être prévisible.
- toute absence de structure ou de régularité.

On trouve que les approches mathématiques gardent le même esprit car à chacune des notions intuitives de l'aléatoire correspond une de ces approches algorithmiques :

- La complexité algorithmique développé par Kolmogorov, Chaitin , parmi d'autres.
- Les martingales effectives développé par Schnorr.
- Les tests statistiques effectifs ou ce qu'on appelle *typicality* en anglais ou 'typicalité' ou 'typicité' en français développé par Martin-Löf.

Ces approches consistent à utiliser la théorie algorithmique de l'information pour définir ce qu'est l'aléatoire. Les auteurs de chaque approche avaient tous dans leur esprit la motivation de capturer ce phénomène. Ultérieurement d'autres applications ont donné une vie propre à la théorie, surtout par rapport à la valeur méta-mathématique que nous exposons plus tard.

1.1.4 Le contenu d'information d'un objet individuel

Grâce au modèle de calcul de Turing, on a un outil puissant pour formaliser le concept de complexité d'une suite par le biais de son contenu d'information.

Pour être prudent et encadrer cette nouvelle théorie de contenu d'information, il faut adopter le concept de calcul pour décrire un objet par un "programme informatique" que produit l'objet (ou sa représentation).

La notion de calculabilité n'étant pas suffisamment précisée auparavant, il a été difficile de donner un cadre stable à la définition algorithmique de l'aléatoire, mais l'utilisation d'un modèle de calcul a donné lieu à plusieurs formalisations.

La solution au phénomène des sorties à pile ou face, dont nous avons parlé dans la première section, a été proposée indépendamment, dans les années 60, par Solomonoff [Solomonoff, 1960], Kolmogorov [Kolmogorov, 1965] et Chaitin [Chaitin, 2001] (le deuxième était déjà à l'origine de la formalisation axiomatique du calcul des probabilités). Si la suite de 000... (0 répété 1 million de fois) nous semble surprenante comme produit du hasard, c'est parce qu'elle est très facile à décrire et non pas difficile comme on l'attendrait pour une suite aléatoire.

Habituellement, la théorie classique de l'information quantifie l'information en demandant combien il faut de bits pour encoder et communiquer un message lors d'une suite de questions oui/non (bits). Il ne faut par exemple qu'un bit pour coder un simple oui ou non. La Théorie Algorithmique de

l'Information (AIT) est d'un caractère très différent de la théorie classique de l'information. Ce qui motive l'AIT, c'est la question : que signifie être aléatoire pour une suite ou séquence de 0 et de 1 ? L'AIT fournit une réponse en se basant sur les longueurs des programmes informatiques décrivant ou reproduisant une séquence. Par exemple, une séquence finie de bits (appelée aussi une chaîne) dotée d'une structure répétitive, peut se décrire de façon plus succincte. En langage familier, on pourrait dire que la chaîne 0101010101 se décrit comme “zéro et un, cinq fois”, alors qu'une chaîne ayant l'air plus aléatoire demanderait une description plus longue, jusqu'à nécessiter peut-être de l'épeler bit par bit.

Pour la théorie classique de l'information, incarnée par l'Entropie informationnelle de Shannon (voir le chapitre sur la Théorie de l'Information) qui s'intéresse, à la base, aux questions d'efficacité et de fiabilité des transmissions, le contenu informationnel d'une séquence est le nombre de bits nécessaires pour quantifier chacune des différentes sous-séquences (de toutes longueurs) contenues dans un message. Contrairement à la Théorie Algorithmique de l'Information, où le contenu informationnel est le nombre minimum de bits nécessaires pour stocker un programme produisant la séquence. On rapproche le concept d'Entropie de Shannon d'un concept tiré d'un champ de la physique, appelé mécanique statistique, dans lequel l'Entropie de Boltzmann est le concept fondateur central. Tout comme celle de Shannon, l'Entropie de Boltzmann décompte le nombre de “micro-états” différents qui constituent un “macro-état” physique. Par exemple, le nombre d'arrangements différents de particules dans une pièce remplie de gaz.

D'un autre côté, la théorie algorithmique de l'information s'appuie solidement sur la théorie de la computation telle qu'initiée par Alan Turing. Développée indépendamment par Ray Solomonoff, Andrei Kolmogorov et Gregory Chaitin avec des contributions influentes de Leonid Levin, Per Martin-Löf, Claus Schnorr, Peter Gács et Charles Bennett, entre autres, l'AIT tente de quantifier des concepts tels que l'aléatoire, la simplicité et la complexité, qui autrement resteraient comme dans le discours courant. Ainsi, l'AIT peut aider à formuler et à s'attaquer à des questions de Philosophie de l'Information, là où les outils traditionnels de la théorie de l'information sont impuissants ou pauvrement équipés pour le faire, tout particulièrement la théorie classique de l'information. l'AIT peut traiter de certaines questions liées à la sémantique et à l'épistémologie. Plus précisément, l'AIT peut apporter un éclairage sur les limites de la connaissance formelle à laquelle on peut mécaniquement ou effectivement accéder (ou non), sur le connu et l'inconnu mais aussi sur le connaissable et l'inconnaissable.

Par exemple, la suite 01101001100101101001011001101001... peut sem-

bler aléatoire selon l'Entropie de Shannon parce que le nombre de 0 et de 1 dans la suite est le même et donc elle a une entropie maximale. Cette suite est simplement générée en partant de 0, puis en accolant successivement le complément booléen (1 quand c'est un 0, et 0 quand c'est 1) de toute la suite obtenue. Un autre regard sur cette chaîne peut mettre en valeur ce processus très simple, 0, 1, 10, 1001, 10010110, 1001011001101001, ... où partant de 0, le prochain bit est 1, et les deux bits suivants sont donc le complément des deux premiers 0 et 1 qui est 10 (1 et 0), les quatre suivants étant le complément des quatre premiers, et ainsi de suite. Malgré la longueur infinie de cette suite, la suite de Thue-Morse, il y a un très court programme de longueur fixe qui peut générer tous les bits (on le décrit en quelques mots du langage courant). Le programme, toutefois, ne sera jamais pris en considération par l'Entropie de Shannon, qui assignera plutôt à cette suite une grande valeur d'entropie informationnelle à cause du nombre croissant de sous-chaînes différentes (alors qu'en réalité, ce programme générateur garde sa même courte taille).

Tout n'est pas chaîne de bits, mais les chaînes de bits sont couramment choisies car elles sont assez simples, mais aussi parce que tout message écrit dans n'importe quel alphabet peut se réécrire en binaire (ce qui ne veut pas dire qu'il est toujours judicieux de le faire). On peut revenir à l'exemple du carré et du gribouillis. Cela a-t-il un sens de se demander quelle forme est la plus simple ? Oui, on peut réfléchir à des logiciels minimums (c'est-à-dire les plus courts) produisant le carré et le gribouillis, puis voir quel est le plus court. La Théorie Algorithmique de l'Information (AIT) va suggérer que le plus court programme représente la forme la plus simple. Nous savons qu'un carré peut facilement se décrire en langage de programmation. En fait, dans la plupart de ces langages, il y a même une fonction spéciale pour dessiner carrés et rectangles, mais cependant écrire cette fonction ne devrait pas être très difficile car un carré peut être déterminé par une caractéristique simple, la hauteur ou la largeur. Au contraire, on peut trouver difficile de dire quoi que ce soit au sujet d'un gribouillis, car probablement la seule façon de le décrire serait de carrément le montrer. Il semble évident qu'on puisse donc s'attendre à ce qu'un gribouillis nécessite un logiciel plus long pour être reconstruit, et donc on pourrait dire de lui qu'il est plus complexe (aléatoire) qu'un carré. Maintenant on peut voir que même si tout n'est pas constitué de chaînes de bits, le carré et le gribouillis ont des programmes qui sont eux-mêmes faits de bits dans un ordinateur, ainsi généralement on va rester dans le monde des bits même si on s'occupe de n'importe quel objet, peu importe que ce soit une image, un graphe ou le plan d'une voiture.

L'importante contribution que Shannon a apportée en découvrant que

l'information pouvait se quantifier (avec le concept fécond de bit), sans considération pour son contenu sémantique (par ex. peu importe que ce soit une voiture ou un vélo), est approché différemment par l'AIT. Contrairement à la supposition courante que les théories quantitatives de l'information ne peuvent apporter aucune lumière quant aux questions liées au sens, certains des concepts avancés par l'AIT sont fortement liés à l'épistémologie. L'un de ces concepts est le concept de compréhension qui a été couplé à plusieurs occasions avec différents concepts d'information algorithmique, notamment la compression de données. On dit aussi que la théorie de l'apprentissage et de la prévision optimale s'est stabilisée dans un certain sens formel grâce à la probabilité algorithmique. Cette dernière est regardée habituellement comme la formalisation d'un paradigme commun en science, connu aussi sous le nom du Rasoir d'Occam, dont une version populaire énonce que entre différentes hypothèses concurrentes, l'hypothèse dotée du plus petit nombre de suppositions doit être sélectionnée, et qui fut au départ posée par William d'Occam comme "Pluralitas non est ponenda sine necessitate" ("la pluralité ne doit pas être posée sans nécessité"). Le nombre Oméga de Chaitin a aussi été appelé le nombre infiniment sage, contenant tout savoir mais dont le contenu est inatteignable.

Il semblerait qu'à comparer la description d'objets afin de déterminer lequel est le plus succinctement décrit, on ait besoin de spécifier une sorte de "langage universel" pour cela, afin de faire sens concrètement, sinon on risquerait de se retrouver avec tel langage dans lequel tel message arbitraire pourrait être codé court quel que soit l'aspect aléatoire qu'il puisse avoir. Par exemple, on pourrait se mettre d'accord pour compresser tout le contenu de l'Encyclopedia Britannica en un seul bit, de sorte que présenté à quelqu'un, ce bit puisse être déplié par cette personne pour redonner toute l'Encyclopedia. Bien sûr, cela n'aurait pas beaucoup de sens de dire que la complexité de quelque chose puisse être totalement réduite si arbitrairement. Ainsi, il faut trouver un langage généraliste pour "décrire" de façon raisonnable tout objet au lieu de quelque chose avec une valeur sémantique trop forte tel que le langage naturel, parce que dans celui-ci on peut accorder un sens très particulier au mots. Ce langage "raisonnable" suggéré par l'AIT, est le langage de programmation des ordinateurs. La raison en est que les logiciels ne font pas que "renommer" les choses, mais il faut que la description en eux reconstruise l'objet original, sans tenir compte du destinataire. Maintenant on peut voir pourquoi l'Encyclopedia Britannica ne peut pas être accouplée avec un "descripteur" court, sans parler d'un bit unique. Ainsi, le logiciel décrivant l'Encyclopedia Britannica devrait être capable d'imprimer réellement tous les tomes dans leur forme originale.

1.2 La complexité algorithmique

La définition de l'aléatoire donnée par la complexité algorithmique fournit des outils pratiques qui permettent son application aux suites finies, à la différence des définitions abstraites et infinitaires de Martin-Löf [Martin-Lof, 1966] ou Schnorr [Schnorr, 1972].

La complexité algorithmique, introduite indépendamment et presque simultanément par Kolmogorov et Chaitin , définit la complexité d'une suite comme la taille du plus court programme qui la génère. La complexité algorithmique d'une suite est alors comprise entre 0 et (à peu près) sa longueur. Plus formellement,

Définition 1 [Kolmogorov, 1965, Chaitin, 2001] : La complexité algorithmique de $s \in \{0, 1\}^*$, relativement à une machine de Turing universelle U , est définie par

$$K_U(s) = \min\{|p| : U(p) = s\}$$

où $|p|$ désigne la longueur du programme p qui engendre s avec la machine universelle U . L'idée principale c'est qu'est simple ce qui se décrit brièvement. Donc, en utilisant la complexité algorithmique, on peut maintenant déterminer que la suite :

0000000000000000000000000000000000000000000000000000000

a une complexité algorithmique plus faible que la suite :

11101001001101110101101001001100011010000001101000

De plus, cette complexité est intrinsèque à la suite et non pas à un ensemble comme l'Entropie de Shannon le mesurerait en donnant une valeur de complexité relative à l'ensemble et non pas indépendante intrinsèque à la suite.

La première peut être produite par un programme du type “imprime 50 zéros” qui est vraisemblablement plus court qu’un programme nécessaire pour créer la deuxième suite.

Donc, en utilisant la complexité algorithmique, on peut aussi déterminer que la suite :

type quelconque de “régularité”, C va s’en servir pour minimiser la longueur du programme informatique. Un algorithme honnête de compression peut se définir comme celui qui transforme une chaîne en deux parties : l’une est la version compressée de l’objet, et l’autre les instructions pour décompresser la chaîne, les deux ensemble comptant pour la longueur finale de l’objet compressé. En d’autres termes, on ajoute l’algorithme de décompression à la chaîne compressée de sorte que la chaîne compressée en vienne à ses propres instructions de décompression. La complexité de Kolmogorov établit la limite ultime de la compression. Elle nous dit pourquoi tels fichiers ne vont pas se compresser au-delà d’une certaine limite, et pourquoi on ne peut pas les compresser à la longueur zéro (à moins que le fichier soit vide).

K peut être très utile pour toutes sortes d’applications liées à la sémantique et la signification, même quand elle traite l’information d’une façon formelle et quantifiable. Imaginons qu’Alice donne à Bob la séquence 01010101 et lui demande s’il croit que cette séquence est le fruit d’un tirage à pile ou face (où 0 est face, et 1 est pile). Bob va-t-il croire que la séquence est honnêtement aléatoire ? Il ne va pas le croire, même si la théorie classique des probabilités dirait que 01010101 possède exactement les mêmes chances d’apparaître que n’importe quelle autre séquence de même taille. Ainsi, la théorie classique des probabilités n’aide pas à expliquer la perception de Bob quant à l’apparence non-aléatoire de la séquence. Qu’en est-il de l’Entropie informationnelle de Shannon ? (voir le chapitre sur la théorie de l’information de Shannon) la théorie de l’Entropie informationnelle de Shannon dirait que 01010101 a une entropie plus basse que, disons, 0110010100, mais l’Entropie de Shannon ne peut pas faire de distinction entre 01010101 et 000000011111, qui peut sembler tout aussi bizarre. En fait, si Alice montre à Bob une séquence telle que 315149... et demande à Bob si cette chaîne lui paraît aléatoire, il y a des chances que Bob reconnaisse les chiffres 315149... comme les chiffres de la constante mathématique π , qui pour Bob ne va pas sembler du tout aléatoire car ils peuvent se générer grâce à un processus pouvant se répéter en reproduisant à chaque fois exactement la même séquence, et donc carrément opposée à ce qu’on pourrait qualifier d’aléatoire. L’Entropie de Shannon, cependant, assignerait une entropie maximum aux chiffres de π , car π est supposé être un nombre normal, nombre qui contient toutes les sous-séquences possibles de chiffres. Par exemple, toutes les sous-séquences de 3 chiffres telles que 123, 234 ou 098 vont apparaître exactement le même nombre de fois dans π , de même pour n’importe quelle autre séquence de n’importe quelle longueur. Ce qui signifie que l’Entropie de Shannon des chiffres de π tend à être une entropie maximale pour les séquences plus longues de chiffres de π . Cependant, π aurait

une complexité de Kolmogorov très basse, car π est le résultat d'un processus algorithmique pouvant être décrit de manière très concise à partir de l'une des plusieurs algorithmes courts qui existent. Pour l'AIT, π n'est pas aléatoire, tout comme Bob le pensait quand il a reconnu que la suite était constituée par les chiffres de π . En pratique, même quand une approximation de $K(\pi)$ va être difficile à estimer grâce à un algorithme de compression, par exemple, K est agencée pour repérer précisément ce type de structure comme étant opposée à une structure aléatoire, contrairement à la théorie classique des probabilités ou à d'autres mesures telle que l'Entropie informationnelle de Shannon. Pour qu'un algorithme de compression reconnaisse que c'est π , il faut qu'il soit suffisamment "futé", et tout comme Bob, qu'il ait une certaine connaissance des mathématiques. Maintenant on peut voir comment l'AIT aide à comprendre la signification, d'une façon formelle, dans ce contexte. La signification pour Bob devant les chiffres de π , veut dire que quelque chose a fait tilt dans la tête de Bob quand il a vu π car il a reconnu que les chiffres avaient un certain sens (au final, cela peut arriver même si Bob l'ignore ou l'a oublié, car π est une relation géométrique précise pour tous les cercles). Le fait que la complexité de Kolmogorov puisse reconnaître π comme un objet spécial comparé à, disons, 0101010101 ou 000000011111 semble saisir l'idée de signification et même de signification subjective, étant donné que l'algorithme de compression, tout comme Bob s'il n'avait pas connaissance de π , aurait échoué à reconnaître π , mais cela ne veut pas dire que Bob ou K n'auraient pas été capables de finalement voir que π est le résultat d'un simple calcul (hautement compressible). L'organisation de π est cachée, mais elle est extrêmement ordonnée et donc algorithmiquement non-aléatoire.

Un autre exemple, c'est la suite nommée Thue-Morse. La suite ci-dessous semble à première vue aussi aléatoire :

0110100110010110100101100110100110010110011010010110100110010110

Cependant, il s'agit de la suite de Thue-Morse construite de la façon suivante :

0

01

0110

01101001

0110100110010110

...

$tm(n+1) = tm(n) \text{ } \langle \rangle \text{ } nottm(n)$ (c'est la concaténation des suites précédentes niées)

Elle vient donc d'un algorithme assez simple (voir court).

1.3 L'aléatoire algorithmique par incompressibilité

Ce n'est qu'avec l'arrivée des travaux de Schnorr et Martin-Löf que la définition de l'aléatoire mathématique prend sa forme actuelle par le biais du concept de martingale [Schnorr, 1972] (stratégie de pari) et de test statistique effectif [Martin-Lof, 1966]. On peut classer les définitions en trois catégories : typicalité, imprédictibilité et incompressibilité.

La notion de typicalité est basée sur la théorie de la mesure, elle formalise l'idée intuitive qu'une séquence binaire infinie satisfait toutes les propriétés statistiques pouvant être testées de façon algorithmique. Une suite s est aléatoire au sens de Martin-Löf [Martin-Lof, 1966], si et seulement si, s a toutes les propriétés communes à la plupart des séquences binaires.

La notion d'imprédictibilité exprime le fait qu'une séquence binaire est aléatoire s'il n'existe aucune façon algorithmique d'en prédire ses bits. Une suite est aléatoire au sens de Schnorr [Schnorr, 1972] si elle est imprévisible et donc aucune stratégie de pari effective ne peut mener à un gain à long terme si l'on parie sur les bits de la séquence. En d'autres mots, une séquence infinie aléatoire doit être une séquence ne possédant aucune structure, régularité, ou règle de prédiction identifiable.

D'autre part, la notion de compressibilité permet de définir la complexité d'une suite infinie. Une séquence est aléatoire au sens de Chaitin-Levin [Chaitin, 2001, Zvonkin and Levin., 1970] si tout segment initial de la séquence est incompressible. La section suivante fournira plus de détails.

Si s est une suite finie alors on dit que :

s est aléatoire si $K(s) > |s| + c$.

L'idée c'est qu'une suite finie est aléatoire, si et seulement si, elle est incompressible.

Si s est une suite (séquence) infinie :

s est c -aléatoire si et seulement si c est une constante telle que pour tout n , la complexité algorithmique des suites $s \upharpoonright n$ est plus grande que $n - c$ où $s \upharpoonright n$ est l'ensemble des préfixes de s de longueur égal ou plus petit que n . s est alors aléatoire si, et seulement si, il existe une constante c telle que s est c -aléatoire.

Le terme c est introduit pour s'assurer que même dès le début pour les segments initiaux et pour les suites courtes la définition est satisfaite. Évidemment, si s est infinie, on ne peut jamais tester la compressibilité de tous les segments initiaux $s \upharpoonright n$ pour tout n et on ne peut donc jamais se prononcer de manière définitive sur le caractère aléatoire de s et le comportement de c par rapport à la taille de s .

Pour déterminer si une suite infinie est aléatoire on utilise donc une généralisation de la définition 1.

Définition 2 [[Kolmogorov, 1965](#), [Chaitin, 2001](#)] : Une séquence infinie est aléatoire si

$$\exists c \forall n, K(s \upharpoonright n) > n - c$$

Autrement dit il y a une constante c pour laquelle pour tout n tous les segments initiaux de s ne peuvent pas être compressés plus de $n - c$ bits. Donc une suite binaire finie s est dite aléatoire si $K(s)$ est proche de la longueur originale de s , autrement dit si s n'est pas compressible. Ce qui est intéressant est que cette définition ne dépende plus d'une machine universelle de Turing en particulier et le "stress" est maintenant sur la petite constante c dont on ne peut rien dire par rapport à sa vitesse de convergence c/n même si on sait qu'à la limite $c/n = 0$ pour $n = \infty$.

D'autre part, si s est une suite finie, elle peut toujours faire partie d'une suite infinie aléatoire s' et donc le caractère non-aléatoire d'une telle suite ne peut jamais être absolu (elle est dans le meilleur des cas "d'apparence aléatoire" ou "random-looking"). Comme on verra donc, il y a une incompatibilité fondamentale entre la définition de l'aléatoire pour des suites finies et la définition de l'aléatoire pour des suites infinies.

Parmi les propriétés des suites binaires, il y a notamment le fait que la majorité des suites finies ont une complexité maximale (c'est-à-dire proche de leur longueur). Par un argument combinatoire, une suite tirée au hasard a toutes les chances d'avoir une complexité algorithmique maximale dès que n est assez grand. Par exemple, parmi toutes les suites de 0 et de 1 de longueur n , pour n fixé :

- moins d’une suite sur 1 024 a une complexité $< n - 10$, c’est-à-dire peut être comprimée de plus de 10 digits ;
- moins d’une suite sur un million a une complexité $< n - 20$, c’est-à-dire peut être comprimée de plus de 20 digits, etc.

L'une des propriétés les plus importantes de cette mesure (pour l'une ou l'autre version) est qu'aucune des deux n'est calculable, c'est-à-dire, qu'il n'existe pas d'algorithme pour calculer la complexité d'une suite quelconque, car aucune procédure ne garantit de trouver le programme le plus court qui l'engendre. Ceci fait douter de trouver des applications pratiques.

Des approximations à ce concept sont, cependant, envisageables. Une façon de contourner le problème de calculabilité est de considérer des approximations calculables. La longueur de la plus courte “description” peut être approchée par sa plus courte forme compressée, en utilisant un algorithme de compression fixé qui repère des régularités.

Il n’y a pas, en général, de méthode effective pour trouver la meilleure compression, mais toute compression possible donne une borne supérieure, et donc la taille de la suite compressée est une approximation à sa complexité algorithmique. Cette approche est utilisée pour des applications pratiques, notamment par Cilibrasi et Vitányi [Li and Vitányi, 2008] pour la classification de données.

1.3.1 Les martingales effectives ou l'aléatoire selon Schnorr

On peut aussi aborder l'aléatoire en d'autres termes. Pensons qu'on enregistre chaque jour et chaque nuit avec un 0 et un 1. On obtient donc une suite de plusieurs milliards de chiffres, un pour chaque jour ou nuit depuis des milliards d'années² jusqu'à aujourd'hui :

$\dots0101010101010101010101010101010101010101010$

Lorsqu'on observe une suite de ce type, on tend toujours à favoriser les chiffres qui maintiennent le motif de la suite. Plus longue est la suite, plus on est certain de gagner le pari. Avec moins de régularités, le pari devient plus risqué. On voudrait capturer ce phénomène de manière formelle.

2. si on croit que cela est arrivé comme cela, sinon on peut commencer depuis la prise de connaissance de chacun.

Dans n'importe quel jeu de pari, à chaque fois que l'on mise, il existe une probabilité p de gagner et une probabilité q de perdre.

Supposons que, comme à pile ou face, celui qui gagne récupère sa mise plus une somme égale à sa mise. Lorsqu'un jeu est en votre faveur (p supérieure à $1/2$), même si vous affrontez un joueur infiniment riche avec une somme de départ finie, vous deviendrez infiniment riche dans une bonne proportion de cas.

Si vous avez une somme de départ infinie, vous deviendrez toujours riche en jouant assez longtemps. Cependant, si p est inférieur à $1/2$ et si vous ne disposez que d'une somme de départ finie, vous perdrez presque toujours.

Par contre, si p est inférieur à $1/2$ et que vous avez une somme infinie de départ vous pouvez gagner avec une stratégie de jeu. Une martingale est une stratégie de jeu et la martingale géométrique, c'est *La martingale* et elle consiste à doubler la mise K à chaque fois qu'on perd jusqu'à ce que l'on gagne.

Lorsque cela arrive après n fois, on a perdu $K + 2K + 4K + \dots + 2^n K$ mais au temps $n+1$ on gagne $2^{n+1}K$ et donc on a gagné à la fin K . On gagne toujours avec une stratégie martingale géométrique. Evidemment pour assurer que cela marche il faut assumer que le joueur commence avec une somme de départ infinie.

Dans le raisonnement des martingales, aucune référence n'est faite aux résultats du jeu à chaque fois. Ce que ces stratégies mettent en évidence, c'est que si le jeu n'a aucun biais, prendre en compte le passé est inutile!

Une séquence s est aléatoire s'il n'existe aucune martingale (effective) gagnante lorsqu'on parie sur le prochain chiffre de s .

Il n'existe aucune martingale M sur une suite s telle que : $\limsup_n M(s \uparrow n) = \infty$.

Où M est le gain (par exemple, combien d'argent on gagne), $s \uparrow n$ les segments initiaux de la séquence s .

L'idée c'est que personne ne peut gagner en pariant en présence d'une suite aléatoire³.

3. Déjà en 1928 von Mises parlait du "principe de l'impossibilité de pari" [Von Mises, 1928]

1.3.2 Les tests statistiques effectifs ou l'aléatoire selon Martin-Löf

L'univers de Cantor dénoté par 2^ω contient toutes les suites binaires infinies (elles peuvent aussi se mettre en correspondance avec les expansions des nombres réels et donc être considérés comme des nombres réels).

Une suite $s \in 2^\omega$ est aléatoire, si et seulement si, s réussit “tous les tests statistiques effectifs”.

On trouve qu'il y a un nombre dénombrable de tests statistiques effectifs⁴ et seulement un ensemble de mesure de Lebesgue nulle de suites ne satisfaisant pas la condition de Martin-Löf. Donc la plupart des suites sont aléatoires au sens de Martin-Löf.

L'idée c'est qu'une suite est aléatoire au sens de Martin-Löf, si et seulement si elle a toutes les propriétés communes à la plupart des suites dans l'univers de Cantor (toutes les suites binaires infinies). En d'autres mots, une suite est aléatoire au sens de Martin-Löf si elle n'a aucune propriété exceptionnelle.

Comme exemples de propriétés communes à la plupart des suites de l'univers de Cantor, on peut citer : être l'écriture binaire d'un nombre transcendant, être normale, suivre la loi des grands nombres, avoir une sous-suite à n chiffres contenant plus de 0 que de 1 ou suivre la loi du logarithme itéré.

Il est important de se rendre compte que la définition de Martin-Löf n'est définie que pour des suites infinies. C'est alors une propriété intrinsèque de cette définition de ne pas permettre d'exhiber une suite aléatoire individuelle.

Propriétés des suites Martin-Löf aléatoires :

- Une suite aléatoire à la Martin-Löf n'est pas calculable, et une suite calculable ne peut pas être aléatoire au sens de Martin-Löf.
- Presque tous les suites sont aléatoires au sens de Martin-Löf, même si ni les suites algébriques ni les suites calculables ne sont aléatoires !

Le consensus autour de la définition de Martin-Löf c'est qu'elle est la bonne définition mathématique de l'aléatoire.

4. On prouve l'existence d'un test universel (auquel on ne peut pas avoir accès) tel que si une suite passe ce test elle passera n'importe quel autre test statistique effective.

1.4 Convergence des définitions

On peut se demander si toutes ces mesures peuvent être prises au sérieux quant à caractériser pleinement les notions intuitives de complexité et d'aléatoire une fois pour toutes. Le résultat le plus surprenant en AIT, c'est ce qui est connu sous le nom de convergence des définitions. C'est un phénomène identique à la convergence des définitions de la notion d'algorithme, qui s'est produit dans les années 1930, quand des gens comme K. Gödel, A. Church, A. Turing, E. Post, S. Kleene, parmi d'autres, ont caractérisé cette notion en usant d'approches différentes et indépendantes, qui se sont toutes avérées équivalentes en puissance computationnelle, cela signifiant que le concept d'algorithme avait été définitivement saisi mathématiquement par toutes ces formulations. Quelque chose de similaire s'est passé avec l'AIT, quand A. Kolmogorov, G. Chaitin, L. Levin, C.P. Schnorr et P. Martin-Löf ont conçu indépendamment différentes approches de l'aléatoire (compression, prévisibilité, "typicité") qui se sont avérées être équivalentes de différentes façons fondamentales, en particulier pour le cas fini (la convergence des définitions pour le cas des séquences infinies est discutable) :

Incompressibilité : Nous avons expliqué la façon dont l'aléatoire peut se caractériser par l'incompressibilité. Selon la complexité de Kolmogorov, si un objet est aléatoire, alors il est impossible de le compresser. La compressibilité est un test suffisant pour le non-aléatoire, cependant l'inverse n'est pas nécessairement vrai car la puissance des algorithmes de compression peut être limitée.

Typicité : Nous avons aussi parlé de la notion de régularité et de la manière de s'en servir pour compresser un objet. L'idée de base est de repérer les endroits où de la régularité apparaît et de les coder avec un marqueur plus court. On peut concevoir des tests statistiques pour les régularités, par exemple une procédure testant si une séquence se compose d'un certain nombre de 1, ou si les digits d'une séquence sont comme les digits d'une constante mathématique en binaire. Les séquences aléatoires peuvent alors être caractérisées comme échouant à tous les tests (calculables) possibles de régularité. En fait, Martin-Löf prouve qu'il existe un test statistique universel unique (mais incalculable) pour toutes les régularités possibles dans une séquence.

Imprévisibilité : Une autre caractérisation de l'aléatoire d'une séquence se fait au moyen de l'imprévisibilité. Schnorr montre qu'il est impossible de gagner de l'argent en devinant la suite des digits d'une séquence aléatoire, quand on utilise une stratégie calculable de pari. On peut sentir intuitivement

que s’il n’y a pas de structures prévisibles et qu’aucune régularité ne peut être repérée dans une séquence aléatoire, on ne peut proposer de stratégie de prédiction pour aucun digit. Si c’est réellement le cas, alors la séquence est dite aléatoire.

La convergence des définitions signifie que chaque définition assigne exactement la même valeur aléatoire que toutes les autres définitions. On peut écrire ce beau résultat de façon courte comme :

$$\text{incompressibilité} \leftrightarrow \text{imprévisibilité} \leftrightarrow \text{typicité}$$

Est incompressible ce qui est imprévisible et ce qui est typique statistiquement (ou non exceptionnel). Une suite de résultats d’universalité (à la fois au sens de général et au sens universel de Turing, ce dernier concept étant une version du premier) [Kirchherr et al., 1997], poussent à conclure que la définition de la complexité aléatoire est mathématiquement objective d’après les résultats suivants :

- Martin-Löf prouve qu’il existe un test statistique universel (mais incalculable) qui vérifie tout ce que vérifient tous les tests statistiques énumérables par le calcul. Sa définition de l’aléatoire est donc suffisamment générale pour englober tous les tests effectifs d’aléatoire.
- Solomonoff et Levin prouvent que la probabilité algorithmique est une stratégie d’apprentissage universelle et optimale, sans connaissance préalable.
- Schnorr prouve qu’une approche par la prévisibilité, basée sur les martingales, conduit à une autre caractérisation de l’aléatoire, qui se trouve être équivalente à celle de Martin-Löf.
- Chaitin prouve qu’une séquence algorithmiquement incompressible est aussi une séquence aléatoire de Martin-Löf.
- La confluence de toutes ces définitions.

Quand cela arrive en mathématiques, on pense avoir objectivement saisi un concept (en l’occurrence l’aléatoire).

Le fait que les notions se révèlent mathématiquement équivalentes [Chaitin, 2001, Schnorr, 1972] alors qu’elles sont fondées sur des notions différentes et conçues indépendamment les unes des autres, conduit à penser que ces définitions réussissent à saisir de façon fondamentale le concept de l’aléatoire. Cependant, certains sont gênés par les arguments de la convergence dans l’AIT, car le nombre de caractérisations équivalentes est beaucoup plus faible que le nombre de caractérisations du concept d’algorithme ou de

computation, et les caractérisations sont peut-être moins solides (prises individuellement) (voir Porter dans la section Références).

Un autre problème à propos des arguments de convergence, a été soulevé par Georg Kreisel, suggérant une possible erreur systématique. En gros, on aurait une collection de définitions qui convergent vers un mauvais développement par rapport à la thèse de Church-Turing, écrit Kreisel, “les résultats d’équivalence ne jouent pas de rôle particulier, simplement parce qu’une bonne raison vaut mieux que 20 mauvaises, qui peuvent bien être toutes équivalentes à cause d’une erreur systématique”. Kreisel semble indiquer que la source de l’erreur systématique puisse être induite, par exemple, par des facteurs historiques, quand un groupe de chercheurs essayent de s’attaquer au même problème, trouvent des solutions similaires qui s’avèrent être équivalentes uniquement parce qu’ils avaient des points de départ et des contextes similaires (ou identiques). Rien ne semble suggérer que ce soit le cas, ni pour la notion d’algorithme (supposée être saisie par le concept de calculabilité), ni pour la notion d’aléatoire (supposée être saisie par l’AIT). Il est toutefois très intéressant d’analyser les fondements de toutes les approches pour ensuite les comparer les unes aux autres, et débattre des implications profondes.

Une fois qu’on obtient que les trois approches algorithmiques de l’aléatoire sont équivalentes, on peut décider d’en approfondir une seule car on sait que ce qu’on déduit d’une approche, on le déduit des autres. Il faut cependant revenir aux autres approches car chacune fournit une interprétation différente du phénomène, quelquefois mises en relief plus par une que par l’autre.

Nous allons exposer quelques particularités de l’approche par le moyen de la complexité algorithmique, qui pourraient être considérées comme des défauts ou des problèmes que la théorie laisse sans réponse définitive.

Que la théorie ne fournisse pas de réponses absolues aux questions sur l’aléatoire peut avoir plusieurs interprétations : soit parce que la réponse de la théorie est précisément qu’il n’y en a pas soit parce que la théorie n’est pas toujours complète et qu’elle ne déploie pas encore tout son pouvoir soit une combinaison de ces deux dernières contraintes, l’une ontologique et l’autre épistémologique.

1.5 La nature subjective de l’aléatoire pour des suites finies

On pourrait convenir d’appeler complexes au sens de la théorie algorithmique, les suites finies telles que : $K(s) > |s|/2$ où $|s|$ est la longueur de

la suite s . Par contre le seuil $|s|/2$ est arbitraire. Il n'y a pas de convention acceptée (même au sens mathématique) pour une telle définition. On impose une phase de transition entre l'aléatoire et ce qui ne l'est pas à décider par un seuil qui ne dépend que d'un choix arbitraire (la place du seuil). Le caractère aléatoire se rend seulement moins subjectif au delà du voisinage du seuil, c'est-à-dire aux extrêmes.

Il faut remarquer (par le raisonnement combinatoire expliqué) que : Une suite tirée au hasard a toutes les chances d'avoir une forte complexité de Kolmogorov. Parmi toutes les suites de 0 et de 1 de longueur n , pour n fixé,

- moins d'une suite sur 1024 a une complexité $< n - 10$, c'est-à-dire peut être comprimée de plus de 10 digits ;
- moins d'une suite sur un million a une complexité $< n - 20$, c'est-à-dire peut être comprimée de plus de 20 digits.
- etc.

Ainsi donc une suite tirée au hasard est très rarement compressible. Le seuil fixé est donc très pertinent pour décider le caractère aléatoire de la plupart des suites, donc celles qui sont facilement distinguables (c'est-à-dire qui sont loin de la portée du voisinage du seuil) ne sont pas très nombreuses. En d'autres termes, pour la plupart des suites (du fait qu'elles ne sont pas assez facilement compressibles) le caractère d'être aléatoire est tout à fait subjectif.

En d'autre mots, si on appelle aléatoires des suites presque incompressibles, l'interprétation précise à donner à "presque" (pour fixer un seuil) dépend d'une appréciation subjective.

1.6 La valeur ontologique et épistémologique des définitions

De la définition de séquence aléatoire, on déduit que toute suite finie est contenue dans n'importe quelle séquence aléatoire infinie, autrement dit, ne pas contenir une suite quelconque serait une propriété exceptionnelle de la suite ce que la définition fournie par Martin-Löf interdit pour une séquence aléatoire. Cela implique que si s n'est pas une suite finie aléatoire définie comme telle par le moyen de la théorie, elle n'est toujours que partie d'une séquence infinie aléatoire ! Et l'inverse : si s est une suite finie aléatoire, elle peut toujours être partie d'une suite plus longue non aléatoire. Toute suite (même non aléatoire) est partie d'une séquence infinie aléatoire (pire encore :

elle est partie de toutes les séquences aléatoires dans chacune une infinité de fois!).

1.6.1 Le problème de l'aléatoire local

Donc les définitions de l'aléatoire pour une suite finie et pour une suite infinie ne sont pas tout fait compatibles, car l'une implique que l'autre ne fait aucune vraie distinction. La définition de séquence infinie aléatoire ne permet pas de discriminer entre les suites finies celles qui sont aléatoires de celles qui ne le sont pas.

En fait, la définition de suite finie aléatoire n'a pas vraiment de sens en ces termes algorithmiques où on laisse coexister une définition pour les suites finies et une autre pour les suites infinies (même si la première peut être décrite en termes de la seconde comme cas particulier). Une définition est une définition quand elle permet d'opposer ce qui la satisfait et ce qui ne le satisfait pas.

1.6.2 Ontologie et épistémologie des définitions de l'aléatoire

D'autre part, du fait que la définition de séquence aléatoire infinie n'apporte rien au sens épistémologique, quand on n'a pas accès ni la connaissance des nombres aléatoires ni aux tests pour les démontrer ou exhiber, c'est la définition de l'aléatoire pour des suites finies qui apporte une valeur épistémologique. Car même si elle n'est pas capable de se prononcer sur le caractère aléatoire absolu d'une suite, elle apporte des outils pour en approcher et donner un sens (même si c'est un sens subjectif) à l'aléatoire en termes de complexité.

De plus, la valeur épistémologique consistant à apporter de la connaissance sur une suite quelconque, soit qu'elle soit aléatoire soit qu'elle ne le soit pas, n'est qu'approchable, car elle n'est pas calculable ; c'est dire, même s'il y a une méthode de calcul, le calcul mené ne garantit pas toujours d'obtenir la réponse correcte (un exemple concret consiste à demander si le programme qui engendre une suite est vraiment le plus court ou non).

D'autre part, “Le caractère aléatoire d'un résultat ne garantit pas qu'il ait été généré par un processus aléatoire et inversement, le caractère aléatoire d'un processus n'implique pas nécessairement qu'il produise un résultat aléatoire”. [Earman, 1986]. En effet, les définitions de l'aléatoire par la théorie

algorithmique de l'information ne fournissent aucune information sur les propriétés du processus générateur, qui peut être intrinsèquement déterministe ou indéterministe.

Le caractère algorithmiquement aléatoire d'une suite finie ne détermine pas le caractère déterministe ou indéterministe du processus qui l'engendre. Néanmoins, la nature de la distribution des sorties, à partir de la fréquence dans laquelle les suites sont produites, fournit une information sur le processus d'origine. Une distribution de ce qu'on attend par la mesure de Levin suggère que le processus générateur conduit des calculs qui ressemblent à ceux qui pourraient être conduits par des programmes sur une machine universelle de Turing (formalisme sur lequel la théorie de la complexité est fondée et définie).

Comme on verra plus tard, c'est la fréquence d'apparition de s , et donc la probabilité $m(s)$, qui détermine si l'origine des suites est un programme de calcul court ou long.

1.6.3 La valeur ontologique et épistémologique des définitions

Étant donné que la théorie moderne du calcul ne fait aucune distinction entre le programme et les données d'entrée du programme, on peut penser aux données fixées d'entrée d'un programme comme faisant partie du programme même, dont la mesure de Levin va prédire la probabilité algorithmique sans s'occuper des entrées. Leonid Levin [Levin, 2003] a démontré qu'il suffit de trouver une fonction "one-way" pour construire un générateur de nombres aléatoires déterministes aussi bon que n'importe quel autre générateur (voir même non-déterministe). Une fonction "one-way" est une fonction facile à calculer mais sa fonction inverse est très difficile à la calculer.

En dépit du manque de valeur ontologique de la définition de l'aléatoire pour des suites finies et de manque de valeur épistémologique de la définition de l'aléatoire pour des suites infinies, car elle ne fournit aucune méthode pour exhiber les suites qu'elle essaie de caractériser, cette définition fournit de la valeur ontologique et ce que j'appelle une valeur méta-épistémologique très riche. D'un côté, parce que la définition de l'aléatoire pour des suites infinies caractérise de manière absolue le caractère aléatoire d'une suite infinie. D'un autre côté, quelques fondateurs de la théorie (notamment Chaitin [Chaitin, 1975]) soutiennent que la valeur de la théorie n'est que méta-mathématique parce qu'elle ne fournit que de la connaissance sur la limite de la connaissance même. Chaitin démontre qu'on peut obtenir des résultats

d'indécidabilité par le biais de la complexité algorithmique (ce qu'on appelle le principe heuristique de Chaitin).

1.7 La définition de la complexité algorithmique est-elle robuste ?

On peut se demander si la définition de la complexité algorithmique, est stable et robuste : Est-ce qu'en changeant de langage (ou de machine universelle de Turing), on ne change pas la complexité mesurée ?

La définition de la complexité de Kolmogorov n'est pas absolument indépendante de la machine U , mais elle l'est à une constante près. Le théorème d'invariance donne sens à la définition de la complexité algorithmique. Ce théorème se démontre en utilisant la possibilité d'écrire un compilateur pour n'importe quelle machine universelle de Turing pour traduire les instructions destinées à une machine pour une autre machine.

En effet, si on utilise un langage de programmation pour mesurer la complexité d'un objet, on voudra maintenir une certaine stabilité à travers les changements de langage de programmation pour $K(s)$ afin d'arriver à une possibilité d'application générale. Un théorème garantit la stabilité de K malgré l'usage de différents langages de programmation (ou différentes machines de Turing universelles). Un moyen de réfléchir sur ce théorème d'invariance, ainsi nommé, est de penser à un programme qui fait la traduction entre deux langages de programmation ou deux machines de Turing U_1 et U_2 (pour les machines de Turing universelles—voir le chapitre sur les machines de Turing et la calculabilité). Car toutes les machines de Turing (particulièrement les machines universelles) peuvent être implémentées sur n'importe quelle autre machine de Turing universelle, la longueur minimum d'un programme sur une machine va seulement voir s'y ajouter la longueur du programme traducteur pour l'autre machine. Pensons à un dictionnaire bilingue entre deux langues. La longueur du dictionnaire est fixe et elle peut dépendre de U_1 et U_2 (les langues dans ce cas), mais une fois décidé quel langage et quel dictionnaire utiliser, la longueur du dictionnaire reste la même quel que soit le mot s du dictionnaire auquel on s'intéresse, peu importe que s soit long ou court. Ce théorème d'invariance est la pierre angulaire qui fonde le domaine de l'AIT, et il est dû à R. Solomonoff (1960, 1964), A. Kolmogorov (1965) et G. Chaitin (1966). Le théorème déclare qu'il existe une constante c qui limite la différence de 2 mesures telles que mesurées sur 2 machines de Turing

différentes (ou deux langages de programmation différents) U_1 et U_2 , ce qui donne pour tout s ,

$$|K_{U_1}(s) - K_{U_2}(s)| \leq c$$

Le théorème d'invariance implique que la différence c entre une mesure de K sous un langage de programmation (ou une machine de Turing) et sous un autre langage de programmation, est constante car c'est la longueur en bits du traducteur (ou compilateur) permettant de passer U_1 à U_2 qui importe, et qui reste constante. Le théorème dit que c peut être relativement grande et peut avoir un impact si s est trop petite (assez petite pour que cela arrive), rendant la mesure de K potentiellement instable pendant un moment. Cependant, dès que s est suffisamment longue par rapport à c , alors l'instabilité de K mesurée sur différents langages de programmation commence à devenir de moins en moins importante, pour finir par converger vers les mêmes valeurs de K . Par exemple, imaginons que sous un langage de programmation U , la chaîne de bits $s = 0101010101$ puisse s'encoder par un programme de longueur 10 bits, ce qui fait $K_U(s) = 10$, alors que pour un langage de programmation T , on ait besoin d'un programme de longueur 15 bits pour reproduire la même chaîne, ce qui fait $K_T(s) = 15$. Si la différence entre $K_U(s)$ et $K_T(s)$ pour tout s est une constante $c = 5$ bits, qui puisse se déterminer en écrivant un traducteur entre U et T , alors cela signifie que pour s , la différence entre $K_U(s)$ et $K_T(s)$ est proportionnellement plus grande que pour une chaîne $s' = 01\dots01$ (100 fois plus longue que s). Ainsi la différence apparaissant entre $K_U(s)$ et $K_T(s)$ pour s' sera relativement plus petite que pour s , car elle va rester autour de $c = 5$ bits.

Malgré cette avancée positive, le théorème laisse une question en suspens. Tandis que ce théorème garantit la convergence des valeurs lorsque la longueur de s augmente, il ne dit rien sur la vitesse ou taux de convergence. En effet, c pourrait être incroyablement grande, rendant le théorème inutilisable pour les chaînes courtes. Applications et récents comptes-rendus suggèrent cependant que c est habituellement assez petite dans la plupart des cas naturels pour être presque toujours négligée. La question philosophique en suspens est alors pourquoi en pratique, c reste petite et qu'est-ce que cela signifie pour un langage informatique ou un calculateur que d'être naturel ou de se comporter naturellement.

Une façon courante d'estimer K est d'utiliser des algorithmes de compression sans perte tels que ceux dont on se sert dans les formats d'ordinateurs communs, comme *gzip* et *rar*. L'utilité des algorithmes de compression sans perte comme méthode d'estimation de K vient du fait que la compression

est un test suffisant d'absence d'aléatoire. La longueur de cette compression sans perte d'un objet s (par ex. une chaîne) est donc une limite supérieure pour $K(s)$, ce qui veut dire que tandis qu'on ne peut jamais savoir si on est arrivé au bout de la compressibilité d'une chaîne, si on arrive à la raccourcir un tant soit peu, on peut au moins dire que sa complexité algorithmique ne pourra pas être plus grande que sa longueur compressée.

Maintenant on voit pourquoi ce débat est important, c'est important car la complexité $K(s)$ de Kolmogorov d'un objet s est liée à la compressibilité que s va avoir, il n'est pas seulement question de savoir si s est aléatoire ou non, mais plutôt de savoir si s est relativement aléatoire ou non, et alors si cette relativisation approche la taille de la constante c , on ne pourra donc pas dire grand-chose de détaillé sur l'aléatoire de cette suite. Mais plus on aura de régularités dans s , plus grand sera le taux de compression. Car la version compressée de s associée avec les instructions de décompression, peut se voir comme un programme compressant s , donc $K(s)$ sera petite si s est hautement compressible. D'un autre côté, si l'objet est aléatoire, cela signifie qu'il n'y a pas de régularités, donc qu'il sera incompressible et possèdera une $K(s)$ de valeur élevée quand s sera qualifié d'aléatoire. Moins c'est compressible, plus c'est aléatoire. Plus généralement, on dit qu'une chaîne est c -incompressible ou c -Kolmogorov aléatoire (ou simplement c -aléatoire) si $K(s) \geq |s| - c$.

1.7.1 Le choix de la machine universelle est important

Alors que la formalisation de la notion d'algorithme dans la théorie de la calculabilité a permis de fixer le modèle de calcul sur lequel on peut définir la complexité algorithmique d'une suite (avec une machine universelle de Turing), sans la relativiser à aucun autre modèle de calcul (ce qui a permis d'encadrer la notion de complexité), la dépendance du langage joue un rôle important lorsqu'on veut évaluer la complexité d'une suite finie, comme le théorème d'invariance le met en évidence.

Pour les suites courtes par exemple, parler de $K(s)$ ne semble pas avoir vraiment de sens, car la constante additive c est trop grande par rapport à la taille des suites courtes. Pourtant, on a envie de dire que *01010101* semble plus simple que *1010110*. On pourrait essayer de les compresser pour approcher la valeur de la complexité algorithmique. C'est cette constante c (le résultat du changement de cadre de référence) qui va être le sujet de réflexion des premiers chapitres de cette thèse. On suggère une manière de contourner expérimentalement cette difficulté par une méthode où la constante additive

devient moins importante, voire négligeable, particulièrement pour les suites courtes. Au moyen d'un calcul massif, nous avons produit des distributions de fréquences pour évaluer la probabilité algorithmique d'une suite et ensuite sa complexité.

Cependant, si vous essayez de compresser une suite aussi courte, vous n'obtiendrez rien. En fait, la suite compressée sera plus longue en général car le compresseur doit inclure des instructions de décompression et parce que les compresseurs ne voient que des régularités statistiques ce qui est plus improbable dans une suite courte.

L'un de mes buts dans ma thèse d'informatique soutenue à Lille en 2011 a été de fournir une méthode alternative aux algorithmes de compression traditionnellement utilisés (mais inutiles pour des suites courtes) et rendre plus stable la définition de la complexité algorithmique. J'ai montré donc que de vraies applications de la théorie algorithmique de l'information sont envisageables.

1.8 Résumé des propriétés de la complexité algorithmique

Une observation tirée de la définition de K est que la plupart des objets sont nécessairement aléatoires et sont donc des chaînes à complexité de Kolmogorov maximale ; nous pouvons nous apercevoir de cela dans l'argument de comptage suivant. D'abord noter que la longueur des programmes pour K est toujours donnée en bits, et les chaînes elles-mêmes sont données en bits. Que ces deux unités soient en bits est commode pour comparer des objets du même type, mais on peut se servir de n'importe quelle autre base ou convention. Néanmoins, on peut prouver que la plupart des objets sont aléatoires de façon maximale, en les reliant à leurs programmes informatiques générateurs. La constante additive c étant supposée donnée, on a 2^n chaînes possibles de bits dotées d'une longueur n exactement⁵, et avec la formule $2^0 + 2^1 + 2^2 + \dots + 2^{n-c} = 2^{n-c+1} - 1$, on a toutes les chaînes de bits possibles dotées d'une longueur jusqu'à $n - c$ bits maximum⁶. Il s'en suit alors qu'il y a considérablement moins de programmes courts que de programmes

5. parce que il y a 2^n chaînes possibles pour une chaîne binaire de longueur n exactement, car c'est le nombre d'arrangements avec répétition avec un alphabet de deux lettres (0 et 1) pouvant constituer des mots de n lettres.

6. Par contre, avec la formule $2^0 + 2^1 + 2^2 + \dots + 2^n$, ou la formule $2^0 + 2^1 + 2^2 + \dots + 2^{n-c}$, on a toutes les chaînes jusqu'à la longueur n maximum ou la longueur $n - c$ maximum. Dans le premier cas, jusqu'à 2^n et $n = 3$, on a 14 chaînes non vides. Dans le deuxième

longs. Donc, le nombre de chaînes de longueur n pouvant constituer une paire avec un programme de c bits plus court s'évanouit exponentiellement. On ne peut pas apparier toutes les chaînes binaires de longueur n avec des programmes binaires de longueur beaucoup plus courte, simplement parce qu'il n'y a pas assez de programmes courts pour encoder toutes les chaînes longues en chaînes plus courtes, même dans des circonstances optimales. En fait, en usant du même argument, il est clair que parmi toutes les chaînes d'une certaine longueur, il va toujours y avoir une chaîne qui ne pourra pas être compressée du tout, pas même par un bit unique. Prenons comme exemple toutes les chaînes jusqu'à la longueur 3, celles-ci étant au nombre de 14 : 0, 1, 00, 01, 10, 11, 000, 001, 010, 101, 100, 011, 110 et 111, pour lesquelles il y a le même nombre de programmes de même longueur, et seulement 6 chaînes de longueur plus courte (0, 1, 00, 01, 10, 11), donc la plupart des chaînes vont être couplées à des programmes de même taille et vont donc avoir une complexité de Kolmogorov maximum. Autrement dit, il n'y a pas assez de programmes en bits de plus courte taille (de longueur inférieure à n) pour se coupler avec toutes les chaînes de bits.

Une autre propriété saillante de K et C est aussi couramment vue comme son plus grand défaut. C'est sa nature incalculable. Une fonction est incalculable s'il n'y a pas de machine de Turing qui garantisse de produire un résultat pour ses entrées, ou en d'autres termes, si la machine calculant la fonction ne s'arrête pas pour un certain nombre de ses entrées. En ce qui concerne la complexité de Kolmogorov, cela signifie que la fonction $s \rightarrow K(s)$ n'est associée à aucune procédure effective (ou machine de Turing), ce qui équivaut à dire qu'il n'existe aucune fonction générale qui, étant donné une chaîne spécifique, va pouvoir générer le plus court programme produisant cette chaîne. Le fait que la complexité de Kolmogorov ne puisse pas se calculer vient du fait qu'on ne peut pas calculer le résultat de tous les programmes, en raison de l'indécidabilité du problème de l'arrêt (Voir le Section sur les machines de Turing et la calculabilité).

Cependant, l'incalculabilité de la fonction $s \rightarrow K(s)$ est aussi la source de sa plus grande force. Contrairement à la croyance courante que le plus grand défaut de K soit son incalculabilité, c'est cette incalculabilité même qui donne à K son grand pouvoir. L'AIT prouve qu'il n'existe aucune mesure calculable pour venir à bout de la tâche consistant à trouver toutes les régula-

cas, jusqu'à 2^{n-c} , on a 6 chaînes non vides. Donc dans les 14 chaînes, on en a 8 longues (de trois lettres) et 6 courtes (de une ou deux lettres). Il y a donc plus de longues que de courtes (on considère court par exemple ce qui est de longueur inférieure à $n - c$ avec $c = 1$ comme cas limite mais le raisonnement vaut même si c est minimal, et a fortiori si c est plus grand). Et l'écart va en grandissant très vite quand n croît.

rités possibles dans toutes les suites infinies possibles. Ceci parce qu'il y a une quantité indénombrable de régularités possibles tandis qu'il n'existe qu'une quantité dénombrable de machines de Turing possibles (ou de programmes informatiques), ainsi elles ne sont pas assez nombreuses pour piéger toutes les régularités possibles. Cependant, il est plus précis de se référer à l'incalculabilité de la fonction $s \rightarrow K(s)$ plutôt en tant que demi-calculabilité, car on peut concrètement approcher $K(s)$ par au-dessus, ce qui veut dire qu'on peut calculer les limites supérieures de K . Une façon habituelle de calculer les limites supérieures de K passe par l'utilisation d'algorithmes de compression. Une limite supérieure triviale de K , pour toute chaîne s , est simplement le programme `print(s)`. Si une chaîne s ne permet aucun autre programme plus court que `print(s)`, alors on dit que s est incompressible ou algorithmiquement aléatoire.

Une preuve de l'incalculabilité de K s'explique parfois en se servant du paradoxe de Berry. Le paradoxe de Berry est un paradoxe de l'autoréférence survenant dans des phrases du genre "considérons le plus petit entier positif non définissable en moins de treize mots". Alors qu'on peut remarquer que l'expression dans la phrase précédente est composée de douze mots, de sorte que ce nombre pourrait être défini en moins de treize mots! Bertrand Russell (1906) fut le premier à étudier ce paradoxe dans la littérature, et il l'attribua à G.G. Berry, un bibliothécaire de la bibliothèque Bodleian à Oxford, qui suggéra un paradoxe semblable mais plus limité. Le paradoxe avec ces expressions est que, puisque tout ensemble de nombres entiers possédant une certaine propriété comprend aussi en lui-même le plus petit entier plus petit que n'importe quel autre possédant aussi cette propriété (par ce qu'on appelle traditionnellement le principe du bon ordre), on devrait être capable de trouver un entier non définissable en moins de treize mots. Cependant l'expression elle-même est de moins de treize mots, et "définit" cet entier; On obtient donc une contradiction. La solution du paradoxe passe par rendre précis le mot "définissable". En ce qui concerne la mesure de K , Chaitin a remplacé le mot 'définissable' par 'pouvant être engendré par un programme informatique', et ce faisant, l'expression avec K ne conduit pas à un paradoxe mais à une démonstration que K ne peut concrètement pas être calculé. Ou bien, tourné d'une autre façon, s'il était possible de calculer la complexité de Kolmogorov de n'importe quelle chaîne, il serait alors possible de générer systématiquement des paradoxes du genre Berry avec K , et qu'il y ait des descriptions toujours plus courtes que ce qu'elles sont supposées être, ôtant toute signification à K . A la fois l'approche de Chaitin et une version de Georges Boolos (1989) utilisant le paradoxe de Berry, conduisent toutes deux à une démonstration du Théorème d'Incomplétude de Gödel d'une façon dif-

férente et plus simple. Cette démonstration peut se résumer informellement par ce que Chaitin appelle le principe heuristique, qui est que les systèmes axiomatiques ne peuvent pas prouver la complexité d'une formule, si cette formule possède une complexité algorithmique plus grande que la complexité algorithmique combinée des axiomes par lesquels on veut prouver la formule.

En résumé :

- $K(s)$ n'est pas calculable. La fonction $s \rightarrow K(s)$ n'est pas récursive (c'est-à-dire pas calculable par algorithme) aucun algorithme ne peut, pour tout suite s qu'on lui fournit en données, calculer en temps fini, la valeur de $K(s)$.
- En pratique, pour évaluer $K(s)$, on utilise des compresseurs (sans pertes) : la taille du fichier comprimé de s par un algorithme C de compression est alors une valeur approchée (une borne supérieur) de $K(s)$.
- La non-calculabilité de $K(s)$ a pour conséquence qu'on ne peut jamais être certain d'être proche de sa valeur (car par exemple une régularité non vue par le compresseur utilisé peut être présente dans s).

1.9 Une variante commode de K

Une question intéressante est de savoir si une chaîne peut souvent être produite par un programme informatique dont les instructions sont tirées uniformément au hasard. Une façon de formuler cette question est de se demander avec quelle probabilité une chaîne peut être produite par une machine de Turing universelle tournant sur un programme aléatoire. Cependant, la somme des probabilités de tous les programmes informatiques pouvant produire une chaîne particulière est supérieure à 1 car il y a un nombre infini de programmes pouvant produire la même chaîne. Le problème est qu'il très facile de générer un nombre infini de programmes en tant qu'extension d'un programme qui produit déjà une chaîne. Prenons le programme qui imprime s , puis imprimons un 1 supplémentaire à la fin, pour l'effacer avant l'arrêt. Ce nouveau programme imprime s mais il n'est qu'une variante spacieuse d'un programme plus simple et qui tronque artificiellement le mot produit par un programme plus complet. Or, pour qu'une mesure soit qualifiée de probabilité il faut que la somme des probabilités additionnées donne 1.

Pour contourner ce problème, Leonid Levin (1974) et Gregory Chaitin (1977) ont conçu une façon de ne considérer que les programmes significatifs. Ce sont les programmes qui ne sont pas des sous-programmes préfixes

d'autres programmes valides. Ces types d'ensembles s'appellent des domaines à préfixe libre, c'est-à-dire, qui n'ont pas pour préfixes d'autres programmes valides. Un exemple classique est l'ensemble de tous les numéros de téléphone. Le seul moyen de joindre une personne en faisant son numéro de téléphone est que ce numéro n'ait pas pour sous-chaîne un autre numéro de téléphone. Si cette personne a comme numéro de téléphone 0123456789, et que quelqu'un d'autre a le numéro 0123, alors la première personne sera injoignable car chaque fois qu'on essaiera 0123456789, la compagnie de téléphone nous renverra vers le 0123. Il y a différentes façons possibles d'éviter cela, par exemple si la compagnie fait appliquer un délai plus long ou un caractère spécial comme indication de la fin d'un numéro. Certains systèmes de banque en ligne, par exemple, utilisent le signe dièse (#) pour indiquer une interruption pour délimiter la suite des chiffres, ce sont ces suites qu'on appelle des suites *autodelimitées*. Comme les programmes informatiques sont aussi des suites, on peut avoir des *programmes autodelimités*, qui sont donc des programmes préfixes, car aucun programme ne peut contenir un signe de terminaison (par ex. le signe dièse #) au milieu. Une façon plus commode de procéder pour le système des numéros de téléphone est simplement de faire en sorte que tous les numéros de téléphone du monde soient de même longueur (incluant le pays et les codes régions). D'un autre côté, si une version plus courte des numéros de téléphone existe localement, le numéro raccourci ne fera jamais partie du segment initial de tout autre numéro de téléphone.

Un programme préfixe est bien un programme dont aucun préfixe n'est un programme valide. Des codages des langages préfixes ont la garantie d'exister pour tout ensemble dénombrable, et l'inégalité dite de Kraft (ou de Kraft-Chaitin) permet garantir qu'en prenant la somme des probabilités des séquences produites par des programmes informatiques, on aura une convergence vers 1, ce qui est la condition nécessaire pour une mesure de probabilité. Pour différencier cette variante de K , nous emploierons un C majuscule. La complexité algorithmique (ou de Kolmogorov-Chaitin) peut maintenant s'écrire $C_U(s) = |p(s)|$. Où U fait maintenant référence à une machine universelle de Turing qui n'accepte que les programmes auto-délimités p , et U s'appelle une machine de Turing universelle à préfixe libre.

La variante C à préfixe libre hérite de la plupart des propriétés de K , telles que son incalculabilité et son invariance, donc nous allons toujours parler de K ou de C sans qu'une distinction explicite soit faite. On ne fera pas donc de vraie distinction entre K et C car elles ne diffèrent que par un terme logarithmique, étant donné que le nombre de bits supplémentaires nécessaires pour délimiter une suite ne représente qu'une petite longueur par rapport (et en fonction) de sa taille. Par convention, quand on parlera de K ,

les assertions s'appliqueront à la fois à K et à C , et quand il sera question de C , cela ne concernera que C .

1.10 Résumé des propriétés de C et K :

- La fonction $s \rightarrow C(s)$ n'est pas calculable. Aucun algorithme ne peut, pour toute suite s dont on fournit les données, calculer en un temps fini la valeur de $C(s)$.
- En pratique, pour évaluer $C(s)$, on utilise des compresseurs (sans perte) : la taille du fichier comprimé de s par un algorithme de compression (sans perte) est alors une valeur approchée (une borne supérieure) de $C(s)$.
- La non-calculabilité de $C(s)$ a pour conséquence qu'on ne peut jamais être certain d'être proche de sa valeur (car, par exemple, une régularité non vue par le compresseur utilisé peut être présente dans s).
- $C(s)$ est robuste au sens où il existe une équivalence à une constante près qui ne dépend pas de l'objet à mesurer, mais de la machine qu'on utilise pour le mesurer. Cette propriété qui donne sens à la mesure est aussi une contrainte pour évaluer vraiment (numériquement) la complexité d'une suite, car cela dépend de la machine choisie, le théorème d'invariance ne garantissant que la convergence asymptotique.

1.11 La probabilité algorithmique et la distribution universelle de Levin

1.11.1 La complexité “préfixe” pour une mesure de probabilité algorithmique

Étant donné que la complexité algorithmique n'est pas suffisante pour définir rigoureusement certains concepts pour lesquels on parlera de choisir des programmes au hasard, il faut utiliser la variante dite “à préfixe libre” de la complexité algorithmique.

Comme on a vu, un code préfixe est un code ayant la particularité de ne posséder aucun mot (suite ou programme) ayant pour préfixe un autre mot. En d'autres termes, aucun mot d'un code préfixe ne peut se prolonger pour donner un autre mot du code.

Cette propriété est souvent recherchée pour les codes à longueur variable, par exemple une suite d'instructions de calcul (par exemple un programme), afin de pouvoir les décoder lorsque plusieurs programmes sont concaténés les uns aux autres. Un code préfixe est un code non ambigu. Par exemple, les codes à taille fixe sont tous des codes préfixes. Tel est le cas des numéros de téléphone.

Une mesure adéquate doit être fondée sur des programmes autodélimités [Chaitin, 2001, Zvonkin and Levin., 1970]. C'est-à-dire où il n'est pas possible de concaténer deux programmes, ce qui rendrait inopérant toute tentative de définition d'une mesure de probabilité, car tout programme qui commence avec un autre programme finirait par contribuer un nombre infini de fois à la probabilité du premier programme, ce qui rendrait la probabilité des programmes divergente et la somme des probabilités de tous les programmes strictement supérieur à 1. On s'arrange donc pour que chaque programme ne soit jamais le début d'un autre (par exemple, si 0001110 est un programme alors 000111011 n'en sera pas un). Un programme délimité est donc un programme qui ne peut pas faire partie d'un autre, car il est limité, par exemple, par une instruction qui signale la fin du programme. Dans plusieurs langages de programmation on utilise des délimiteurs. Par exemple, on dénote la fin des instructions par point-virgule, accolades ou le mot *end*.

Pour tirer un programme au hasard, on procède de la manière suivante : on convient d'écrire les programmes en langage binaire et on exige que chaque programme contienne une suite unique pour délimiter la fin du programme. Le tirage au hasard d'un programme consiste à choisir, par pile ou face avec une pièce non truquée, des 0 et des 1 jusqu'à avoir un programme complet, et ensuite à le faire fonctionner. Si la suite de tirages de 0 et de 1 ne donne jamais un programme complet, on considère qu'il n'y a pas arrêt.

1.11.2 Prévisibilité optimale, probabilité de Solomonoff et la demi-mesure de Levin

Vous ne pariez pas pour “le soleil ne se lèvera pas demain” parce qu’obtenir un 1 après la suite . . . 01010101010 est algorithmiquement plus probable.

Solomonoff [Solomonoff, 1960] introduit un concept d'inférence algorithmique que Levin [Zvonkin and Levin., 1970] formalise avec la demi-mesure⁷

7. *demi* indique que la mesure n'est qu'à moitié calculable. Autrement dit, on peut seulement s'approcher de $m(s)$, car on ne peut pas vraiment la calculer et donc la considérer comme une mesure pleine.

qu'il développe mathématiquement dans son concept de recherche universelle. La probabilité algorithmique dit qu'un bit isolé n'est en rien la chaîne aléatoire la plus complexe, mais plutôt la plus structurée possible et, chose plus importante et plus en accord avec l'intuition, que la transition vers la complexité est régulière.

Il se peut que cela fasse sens qu'un bit isolé puisse être regardé à la fois comme la plus simple et la plus complexe des chaînes selon différents points de vue, mais l'approche par la probabilité algorithmique est qu'elle fournit non seulement une notion différente de la complexité d'un bit isolé, notion plus en prise avec l'intuition, mais aussi elle va générer un résultat différent de l'approche par la compressibilité, même quand les deux mesures sont intimement liées et produisent asymptotiquement les mêmes résultats sur le long terme (pour les chaînes plus longues). Ces deux vues reflètent différents aspects de ce que représente un bit isolé.

Il existe une mesure m décrivant la probabilité pour qu'une machine de Turing produise une suite (ou chaîne) s lorsqu'elle tourne selon un programme produit au hasard. m donne une distribution couvrant l'ensemble de toutes les suites, qui est connue sous le nom de Distribution Universelle, dont les propriétés ont même été décrites comme 'miraculeuses' dans la littérature, étant donné son extraordinaire puissance. L'idée cachée derrière m est intuitive et puissante. Si on voulait sortir les décimales de π par hasard, il faudrait essayer encore et encore jusqu'à réussir à tomber juste sur les premiers chiffres correspondant au segment initial du développement décimal de π . La probabilité de succès est extrêmement réduite : $1/10$ par digit multiplié par la quantité désirée de digits. Par exemple, $(1/10)^{2400}$ pour un segment de 2400 digits de π . Mais si, contrairement à un tirage de nombres au hasard, on tirait des programmes au hasard à lancer sur un ordinateur, le résultat serait très différent. Un programme produisant les chiffres de π aurait une probabilité plus importante d'être produit. Des formules concises et connues de production de π pourraient apparaître et être implémentés comme programmes courts produisant un nombre arbitraire de décimales de π . Sous la forme d'une équation, cette mesure de probabilité peut s'écrire comme suit :

Définition 4 [Zvonkin and Levin., 1970] $m(s) = \sum_{p \text{ s'arrête et produit la suite } s} 2^{-|p|}$ avec p un programme autodélimité.

Où $|p|$ est la longueur (en bits) des programmes produisant la chaîne s , et tournant sur une machine de Turing universelle U . Pour pouvoir fonctionner,

U doit remplir un minimum de conditions techniques, c-à-d qu'un logiciel valide ne doit pas être le commencement d'un autre logiciel valide. Il faut remarquer que le terme le plus grand pour la somme de l'équation s'obtient quand le dénominateur est le plus petit, donc quand $|p|$ est le plus petit, nommément la plus courte longueur possible du programme p en bits qui produise s . Mais la longueur du plus court programme n'est rien d'autre que $C(s)$, donc nous avons trouvé une belle relation précise entre fréquence de production d'une chaîne et sa complexité algorithmique.

La mesure m (qualifiée de distribution universelle miraculeuse par Ming Li et Walter Kirchherr [[Kirchherr et al., 1997](#)]) indique que plus une suite s est simple, plus sa probabilité $m(s)$ comme résultat d'un calcul, est grande. Plus une suite s est aléatoire, plus sa probabilité $m(s)$ est petite. $m(s)$ est une distribution de fréquences qui donne la probabilité pour une machine de Turing M de produire une séquence s avec un programme aléatoire. Cette distribution est liée à la complexité algorithmique $C(s)$, car $m(s) = 1/2^{C(s)+c}$ [[Zvonkin and Levin., 1970](#)] où c est une constante.

Le théorème du codage est un théorème reliant la probabilité algorithmique à la complexité algorithmique. La probabilité algorithmique est liée à la complexité algorithmique en ce que $m(s)$ est au moins le terme maximum dans la sommation des programmes, c'est-à-dire celui correspondant au plus court programme, qui a le plus grand poids dans la sommation des fractions définissant $m(s)$.

Les implications dans le monde réel sont larges et fascinantes, si on pose l'hypothèse d'un monde soumis à un processus algorithmique qui se déploie. Cette approche formalise le principe connu du Rasoir d'Occam, couramment interprété comme la recherche de l'explication la plus simple pour une observation. Ici l'objet observé est la suite s , et sa plus simple "explication" est le plus petit programme informatique qui produit s parmi tous les programmes possibles qui engendrent s .

En l'occurrence, si tous les phénomènes naturels peuvent s'exécuter comme les computations d'une machine de Turing, alors $m(s)$ serait la probabilité algorithmique de l'apparition concrète d'un événement s . En fait, $m(s)$ peut servir à expliquer l'apparition de structures à partir d'un aléatoire (de programmes aléatoires), à la plus petite échelle, faisant apparaître l'ordre dans l'univers à partir de rien.

1.11.3 De la métaphore du singe dactylographe au singe programmeur

Pour expliquer le concept de la probabilité algorithmique on peut utiliser une métaphore, c'est la métaphore du singe dactylographe que en sa version originale a été présentée par Borel [Borel, 1913]. Cette fiction ancienne suggère que avec assez de temps, un singe qui tape (on admet que chaque événement est indépendant, c'est-à-dire que le singe tape vraiment de manière aléatoire) indéfiniment sur une machine à écrire dactylographiera n'importe quel texte (e.g. le roman "Les Misérables"). La probabilité d'obtenir Les Misérables serait donc $1/50^n$, avec n la taille de l'oeuvre de Victor Hugo et 50 le nombre de touches dans la machine à écrire.

Cependant, dans la métaphore moderne si on remplace la machine à écrire par un ordinateur [Chaitin, 1966], la probabilité ($m(s)$) de produire le roman des Misérables est beaucoup plus forte (même si elle reste toujours petite car la longueur du texte est assez grande), car la complexité $C(s)$ du texte est très faible par rapport à une suite aléatoire de la même taille.

Le singe qui est une métaphore d'une source de données ou programmes aléatoires devient une analogie d'une source de programmes aléatoires. Si on tire des programmes au hasard on verra que la plupart de ces programmes ne produisent pas de séquences aléatoires.

Plus encore : le singe pourrait suivre une énumération effective des touches et le résultat de $m(s)$ ne variera pas, tandis que s'il suit une énumération dans le cas de la machine à écrire évidemment il ne produira le roman que si l'énumération est exactement celle de l'ordre des lettres dans le roman des Misérables. En d'autres termes, le réquisit d'une source aléatoire est plus faible pour la mesure m (on pourrait remplacer sans problème le singe "aléatoire" par une source déterministe).

Autrement dit, le singe a beaucoup plus de chances de produire un roman en tapant un programme sur le clavier d'un ordinateur que sur une machine à écrire. Le singe, n'est qu'une source de programmes aléatoires. La métaphore, formalisée par le concept de probabilité algorithmique (ou demi-mesure de Levin), indique que si on tire des programmes au hasard, on verra que la plupart de ces programmes ne produisent pas de séquences aléatoires mais plutôt organisées.

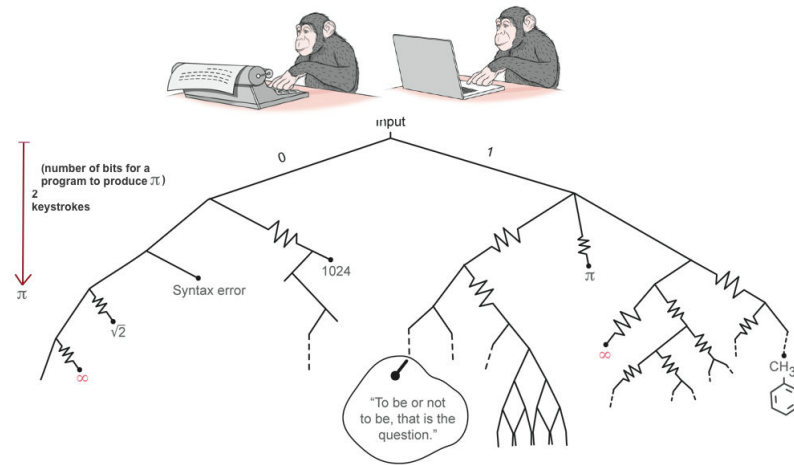


FIGURE 1.1 – D’après la probabilité algorithmique, un singe aura beaucoup plus de chance de produire un objet structuré en le plaçant sur un ordinateur (ou chaque touche représente un programme) que sur une machine à écrire (ou chaque touche est une donnée).

1.11.4 Complexité et fréquence

La théorie des probabilités algorithmiques nous dit qu’une chaîne de basse complexité, par ex. la répétition de “01” une centaine de fois, va être produite par un grand nombre de programmes informatiques tirés au hasard bit par bit, selon une certaine probabilité informatique. Et d’après le théorème du Codage, parmi les programmes qui produisent la chaîne, il va y avoir un très court programme parmi les programmes qui la produisent, court par rapport à la longueur de la chaîne elle-même, car il va faire croître la chaîne en répétant la même structure. Cela peut s’interpréter de la façon suivante. Si on se présente avec une chaîne supposée être toujours croissante à partir d’une source en se demandant quels digits vont sortir après la répétition cent fois de “01”, la réponse sera déterminée selon une probabilité algorithmique par les bits qui vont préserver la complexité de Kolmogorov d’origine (basse) de cette chaîne dotée de cent répétitions de “01”. Dans ce cas, la réponse est que la probabilité algorithmique va nous dire que le motif “01” va se répéter encore et encore, car c’est le motif qui préserve le plus la longueur d’origine du programme informatique générateur. Donc, tandis que la chaîne “01” cent fois va croître très vite, le programme générateur va seulement très peu grandir, d’environ $\log 2n$. Maintenant supposons que 0 représente la journée et 1 représente la nuit, on peut encoder les jours d’une année comme une séquence de 365 digits répétant le motif 01, si on ne vit pas près des pôles. Maintenant

on peut formaliser la question de savoir si la suite de 01 va continuer à se répéter après le 365ème jour. La probabilité classique seule dirait que n'importe quoi peut arriver (qu'une nuit peut venir après une nuit sans qu'on voit la lumière du soleil, au cours de l'année suivante, ou que 10 journées peuvent se succéder suivies de 3 nuits consécutives). Au contraire, on voit jour après jour un motif répétitif dû au mouvement de la Terre par rapport au Soleil, qui va plus probablement se répéter que s'arrêter. Les moyens de compléter la théorie classique des probabilités sont traditionnellement de nature bayésiennes, en attribuant un certain poids pondéré aux observations précédentes. On peut plus naturellement justifier un motif répétitif et une attente au moyen de la probabilité algorithmique. Dans ce cas, le programme derrière une séquence de ce type obéit aux lois de la gravitation décrivant les règles du mouvement qui établit la régularité de la Terre par rapport au Soleil. La probabilité algorithmique est en ce sens un outil optimal de prévision qui montre comment cela se reflète dans les mesures, et comment essayer d'estimer numériquement des valeurs pouvant fournir des applications utiles et une aide à la compréhension. Une autre application de la complexité algorithmique peut se retrouver dans les tests d'aléatoire subjectif, qui testent la capacité humaine à percevoir et à fabriquer de l'aléatoire. Par exemple, la plupart des humains éduqués vont favoriser des séquences contenant des nombres impairs et des nombres premiers, car ils croient que peut-être, les nombres pairs semblent moins aléatoires. La théorie classique des probabilités dirait que c'est faux, et que nombres impairs, pairs, premiers n'ont rien de spécial les uns par rapport aux autres. La probabilité algorithmique dirait que produire tel type de nombre est moins aléatoire et pourrait ainsi mieux quantifier la faute. Car il se trouve que les humains sont de piètres créateurs de nombres aléatoires dans les deux cas, mais dont les erreurs caractéristiques ne pouvaient pas être objectivement quantifiées auparavant.

Nous avons vu comment $m(s)$ est relié à $C(s)$, étant donné que selon sa définition, m obtient le plus grand terme de la somme à partir du plus court programme produisant s . le théorème du Codage formalise ensuite la relation inverse entre $C(s)$ et $m(s)$, en établissant que $C(s)$ vaut environ $-\log m(s)$. Ce théorème indique que la complexité algorithmique d'une chaîne s est très proche (à la constante additive près) de la valeur négative du logarithme de la fréquence de s , il nous dit que si une chaîne s est produite par de nombreux programmes, alors il existe aussi un programme court produisant s . Ce n'est pas complètement par hasard que cette relation ressemble à la formule d'Entropie informationnelle de Shannon, mais contrairement à celle de Shannon, cette autre mesure est liée à une véritable mesure de complexité au sens où elle permet de saisir toute régularité en s , ayant hérité de la puissance

de K , même si cela signifie que K est plus difficile à estimer que l'Entropie de Shannon. Une question de grande importance est quelle distribution de programmes informatiques est supposée pour calculer $m(s)$. Une propriété sympathique de $m(s)$ et la raison pour laquelle on s'y réfère souvent comme la distribution Universelle des probabilités, vient du fait qu'on peut prouver que $m(s)$ domine toute autre mesure de probabilité des programmes informatiques. C'est quelque chose de semblable au théorème d'invariance, signifiant que si une autre probabilité algorithmique $m'(s)$ est proposée par un certain autre moyen, la nouvelle mesure va finir par se comporter exactement comme $m(s)$ l'aurait fait, et donc la fréquence des suites produites par des moyens algorithmiques (par exemple avec une machine universelles de Turing) sera toujours la même à la limite, c'est-à-dire la distribution de fréquences sera exactement la même pour n'importe quelle procédure algorithmique (ou autre machine universelle de Turing). La mesure $m(s)$, toutefois, suppose que la distribution des programmes tirés au hasard soit uniforme, ce qui veut dire que toutes les instructions dans le programme aient les mêmes chances d'apparaître. Cette supposition est source de critiques, car elle constitue une supposition a priori qui peut se voir comme étant du même type que les approches bayésiennes traditionnelles, en invoquant des distributions précédentes non-informatives (de quoi une distribution à l'air ou n'a pas l'air), ce qui demande des précautions.

C'est de cette façon que $m(s)$ et le théorème du Codage reconnectent $K(s)$ à la théorie classique des probabilités grâce à la question de la distribution des programmes, à partir de la supposition que des programmes de même longueur sont équiprobables, laquelle peut être critiquée en disant que dans les situations réelles, ce n'est peut-être pas le cas. Pour défendre l'approche, cependant, on peut faire valoir qu'une distribution uniforme est la distribution non-informative la plus simple selon le principe d'indifférence, qui est que si n possibilités sont indiscernables, alors chaque possibilité devrait se voir accorder une probabilité égale à $1/n$.

1.12 Le scepticisme sur la nature (in)déterministe du processus générateur

“Le caractère aléatoire d'un résultat ne garantit pas qu'il ait été généré par un processus aléatoire et, à l'inverse, le caractère aléatoire d'un processus n'implique pas nécessairement qu'il

produise un résultat aléatoire”. J. Earman, 1986. [[Earman, 1986](#)]

Comme on l’avait mentionné auparavant, les définitions de l’aléatoire par la théorie algorithmique de l’information ne fournissent aucune information sur les propriétés du processus à l’origine d’une suite, processus qui peut être intrinsèquement déterministe ou indéterministe. Le caractère algorithmiquement aléatoire d’une suite finie ne détermine pas le caractère déterministe ou déterministe du processus qui l’engendre.

C’est la fréquence d’apparition de s dans les sorties d’un processus générateur, et donc la probabilité $m(s)$, qui détermine si l’origine des suites est un programme de calcul de la même façon que trouver une distribution gaussienne suggérerait fortement que le processus d’origine a été un processus aléatoire au sens classique.

Étant donné que la théorie moderne du calcul ne fait aucune distinction entre le programme et les données d’entrée du programme, on peut regarder les données fixées d’entrée d’un programme comme faisant partie du programme même, dont la mesure de Levin va prédire la même probabilité algorithmique sans s’occuper des entrées. Leonid Levin [[Levin, 2003](#)] a démontré qu’il suffit de trouver une fonction “one-way” pour construire un générateur de nombres aléatoires déterministes aussi bon que n’importe quel autre générateur (voir même non-déterministe).

Comme on l’a expliqué auparavant (section de la demi-mesure m), la probabilité algorithmique n’a pas besoin nécessairement d’une source aléatoire car la plupart des énumérations effectives (et évidemment déterministes) produiront les mêmes résultats.

Dans une étude statistique sur les sorties des générateurs des nombres aléatoires on a trouvé [[Calude et al., 2002](#), [Calude and Stay, 2008](#), [Delahaye and Zenil, 2012](#)] que pour les tests statistiques les plus connus (une centaine) appliqués à des suites (pseudo) aléatoires d’origines bien déterministes et (dites) non-déterministes (générateurs basés sur des processus quantiques) on pouvait repérer peu de différences. En fait, ce sont 2 suites qui ont le mieux réussi (elles ont passée tous les tests statistiques), il s’agit d’une suite produite par le générateur des nombres aléatoires du logiciel *Mathematica* et la suite des décimales du nombre π

1.12.1 Vers une définition stable ou comment tester la nature algorithmique du monde

Si on fait fonctionner une machine universelle (à programmes auto-délimités) en lui donnant un programme au hasard, alors elle produit la suite s avec la probabilité $m(s)$ (la mesure de Levin [Zvonkin and Levin., 1970]). Alors calculer $m(s)$ pourrait nous approcher de $K(s)$ par la formule :

$$m(s) = 1/2^{K(s)+c}$$

Les expériences suggèrent qu'on peut donner un sens stable à $m(s)$, même pour des s petites en procédant expérimentalement :

- en faisant fonctionner des mécanismes de calculs de manière systématique (machines de Turing, automates cellulaires, etc.) pour produire des suites s au hasard ;
- ou en prenant des données du monde réel et en les traduisant en suites de 0 et de 1 ; puis en observant quelles sont les distributions obtenues pour les s .

Ce travail a été fait par Delahaye et Zenil [Delahaye and Zenil, 2007].

1.12.2 Convergence en distribution

Tout dans la théorie de la complexité aléatoire de Kolmogorov et Chaitin (y compris la théorie de la probabilité algorithmique de Solomonoff et Levin) est défini de manière asymptotique, c'est-à-dire, pour des suites de longueur croissant vers l'infini. En effet, par exemple le théorème d'invariance qui garantit qu'une machine universelle de Turing va finir par produire les valeurs de $K(s)$ ou $C(s)$ s'appuie sur cet argument asymptotique, de même pour la probabilité algorithmique $m(s)$. Par contre, si ses résultats donnent bien naissance à ses théories, ils sont des résultats plutôt négatifs car les théorèmes ne garantissent pas le taux de convergence et donc on ne peut jamais savoir si une approximation de $K(s)$, $C(s)$ ou $m(s)$ est proche ou pas des valeurs réelles. Par convergence on veut dire donc, la convergence d'une valeur vers sa valeur dite réelle ou théorique selon les définitions asymptotiques.

On peut, donc, étudier le comportement des distributions expérimentales pour évaluer la stabilité des classements produits. Nous croyons avoir trouvé une distribution distinguée m_e que l'on conjecture égale à la distribution générée à partir de la mesure m .

TM(2,2)		CA(1)	
string, frequency		string, frequency	
000, 0.436	————	000, 0.016	
111, 0.435	————	111, 0.012	
010, 0.046	————	010, 0.012	
101, 0.046	————	101, 0.010	
100, 0.014	————	100, 0.005	
011, 0.014	————	001, 0.004	
110, 0.003	————	011, 0.003	
001, 0.003	————	110, 0.002	

FIGURE 1.2 – Comparaison des distributions de fréquences des sorties : TM et CA.

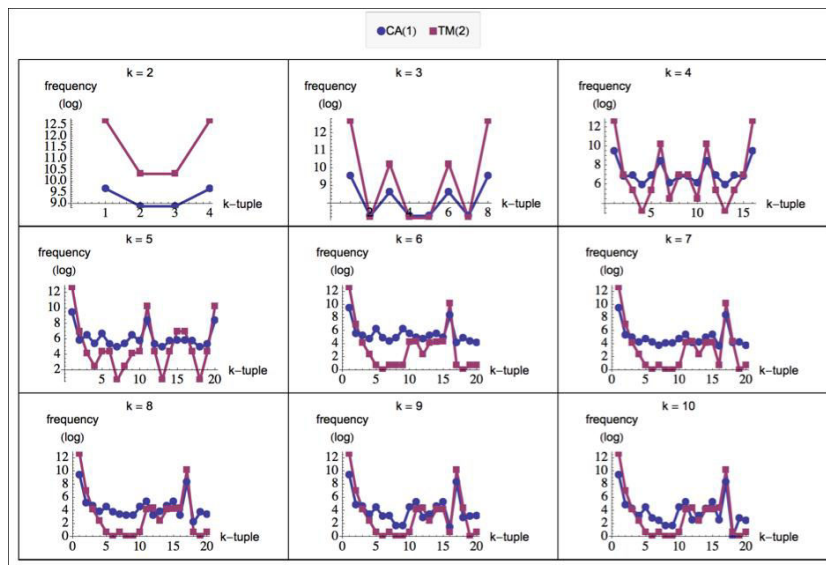


FIGURE 1.3 – Graphique de corrélation des distributions des fréquences des sorties : TM et CA.

On propose alors une distribution naturelle, et on propose de définir des machines de Turing comme naturelles et comme c -exactes si elles s'approchent de la distribution naturelle. Une machine universelle de Turing est naturelle si elle préserve depuis le début (donc même pour les courtes suites) l'ordre des suites imposé par la distribution naturelle. Une machine universelle de Turing est c -exacte si elle ne s'écarte jamais par plus d'une distance c des valeurs de la distribution naturelle. Il semble donc possible de définir m (et donc K) d'une manière indépendante du langage. Ce qui rend stable la définition de la complexité algorithmique.

1.12.3 Le nombre à la sagesse infinie : le nombre Omega (Ω) de Chaitin

Nous allons brièvement nous déplacer vers le sous-domaine de l'AIT qui étudie l'aléatoire algorithmique des objets infinis, où très peu, sinon aucune application réelle n'a été développée, mais où un riche débat fondamental a surgi. La probabilité algorithmique en association avec la fonction de distribution universelle, $s \rightarrow m(s)$, sont toutes deux étroitement liées à un autre concept fertile dans la théorie algorithmique de l'information, qui est la probabilité de l'arrêt selon Chaitin [Chaitin, 1975], aussi connue comme l' Ω (Oméga) de Chaitin, nombre défini par $\Omega_U = \sum_{p \text{ s'arrête}} 1/2^{|p|}$. Dans cette formule, tous les programmes p tournant sur une machine de Turing U (à "préfixe libre" ou "sans préfixe") et qui finissent par s'arrêter seuls, contribuent à la somme des valeurs $1/2^{|p|}$. Plus le programme p sera long⁸, plus petite sera la valeur $1/2^{|p|}$, et donc plus petite sa contribution à Ω . Les programmes courts, au contraire, contribuent largement aux valeurs les plus significatives pour un nombre Ω . Tout comme K et C , Ω est lui aussi demi-calculable, ce qui signifie qu'on peut l'estimer (dans ce cas par dessus) en fixant un cadre de langage informatique en faisant tourner des programmes aléatoires.

Le problème de l'Arrêt et l' Ω de Chaitin : Comme c'est bien connu, le problème de l'Arrêt pour les machines de Turing est le problème de décider si une machine arbitraire de Turing va finir par s'arrêter quand elle est alimentée

8. Un programme qui ne s'arrête pas, pourrait peut-être s'arrêter à un moment, mais plus loin que ce qu'on a été voir pour l'instant. On n'a pas eu le temps d'y aller voir, car le programme continue trop longtemps sans s'arrêter. Son arrêt reste hors de notre portée, même s'il existe. Cela n'a pas d'importance s'il va s'arrêter ou pas, on ne le saura peut-être jamais. Pour demi-calculer Omega, on ne tient compte que des programmes connus pour s'arrêter, et plus le programme sera court, plus il contribuera à la valeur d' Ω (Omega), c'est la règle de construction d' Ω .

par une entrée arbitraire s . Les computations d'arrêt peuvent être reconnues simplement en faisant tourner les machines le temps nécessaire pour qu'elles s'arrêtent. Le problème est de détecter les programmes qui ne s'arrêtent pas, pour lesquels on ne peut pas savoir si le programme va continuer à tourner sans arrêt ou va finir par stopper. Une représentation élégante et concise du problème de l'arrêt est le nombre irrationnel Ω de Chaitin, défini comme la probabilité d'arrêt d'un calculateur universel programmé par des tirages pile-ou-face.

Une machine universelle, c'est-à-dire susceptible de représenter toute fonction calculable, étant donnée, on numérote les programmes par ordre de longueur, puis par ordre "alphabétique" (0 puis 1) à l'intérieur des programmes de même longueur. Soit $p_0, p_1, \dots, p_n, \dots$ une telle numérotation.

Chaque programme, une fois lancé, finit par s'arrêter ou, au contraire, poursuit indéfiniment ses calculs (par exemple, parce qu'il boucle). La définition du nombre Ω de Chaitin fait intervenir le problème de l'arrêt de la manière suivante.

Définition 3 [[Chaitin, 2008](#)] : Ω est la probabilité qu'un programme p tiré au hasard s'arrête : $\Omega = \sum_{p \text{ s'arrête}} 2^{-|p|}$ avec p un programme autodélimité.

De la définition de Ω , on sait tirer toutes sortes d'informations et démontrer ses propriétés (il est transcendant, équiréparti, etc.). Mais justement, l'une de ces propriétés signifie qu'on ne pourra jamais connaître tous ses bits, qui se comportent comme une suite de tirages aléatoires totalement imprévisibles.

Connaître les premiers n bits d' Ω devrait nous permettre de décider si, oui ou non, chaque programme jusqu'à n bits de long va s'arrêter, donc connaître la totalité des digits d' Ω nous permettrait de sortir de l'indécidabilité du problème de l'arrêt pour tous les programmes possibles. C'est pourquoi Ω peut se voir comme l'encodage de toutes les réponses possibles à n'importe quelle question calculable. Etant donné qu'on peut toujours formuler les questions en termes de savoir si une machine de Turing va s'arrêter (réponses oui et non), on devrait avoir des réponses à toutes les questions mathématiques. On peut penser au nombre Ω comme à un oracle rappelant la réponse de l'ordinateur 'Pensée Profonde' à la Question Ultime de la Vie, de l'Univers et du Tout dans le livre de science-fiction de Douglas Adams *Le Guide Galactique*. Mais tout comme dans cette histoire de science-fiction où l'ordinateur donne

comme réponse le nombre 42, les réponses données par un nombre Ω seraient difficiles à comprendre et, en principe, impossible de s'y conformer.

L' Ω de Chaitin est en fait une famille de nombres, car son développement digital dépend de la machine de Turing universelle U choisie ou du langage de programmation. Pour chaque U choisie, il y a un nombre Ω différent. Comme une mesure de probabilité, la somme des termes dans Ω ne doit pas dépasser 1 et donc U doit aussi être une machine de Turing à préfixe libre (ou au langage de programmation auto-délimité), comme c'est aussi le cas pour $m(s)$. Parce qu' Ω ne peut jamais être 1, et étant donné qu'un certain nombre de machines de Turing ne vont jamais s'arrêter, Ω s'appelle plus précisément une mesure de demi-probabilité (tout comme $m(s)$).

En effet, la mesure de demi-probabilité de Solomonoff-Levin (ou juste demi-mesure) $m_U(s)$ fournit une approximation de Ω_U , pour le même U , ainsi que la valeur de la fréquence des chaînes produites par les programmes aléatoires qui s'arrêtent, et d'après laquelle leur probabilité algorithmique de chaîne (et la complexité algorithmique $K(s)$ de Kolmogorov pour les chaînes, d'après le théorème du Codage) peut être estimée. Donc tout comme $m(s)$, le nombre Ω est demi-calculable par dessous, car on peut numériquement estimer les limites les plus basses des deux mesures. Les mesures demi-calculables comme Ω et $m(s)$ sont aussi appelées énumérables par calcul (souvent cité en raccourci par c.e. ou r.e.).

Quand on pense à Ω en terme de nombre sage contenant une connaissance infinie, incluant les réponses à toutes les questions pouvant être formulées comme des programmes de calcul (par ex. tous les problèmes mathématiques ouverts et plus), il est très intéressant d'apprendre que les digits d' Ω sont inatteignables et incompressibles, cela signifiant qu'il n'existe aucun accès à cette connaissance et qu'il n'y a aucun raccourci pour l'atteindre. Aucun processus ne peut dépasser Ω , car il ne peut se déduire par aucun moyen plus simple que la séquence elle-même d' Ω en bits. Cela ne veut pas dire qu'on ne peut pas calculer quelques digits d' Ω dans de nombreux cas. Par exemple, si nous savions que les programmes 0, 10 et 110 s'arrêtent tous (à noter qu'ils sont à préfixe libre), alors on saurait que les premiers digits d' Ω sont 0.111, et inversement, si on était parti de 0.111 d'après ce nombre Ω , on saurait que les programmes 0, 10, 110 s'arrêtent. Dans cette optique, Ω encode et "compresse" de façon maximale l'information de l'état d'arrêt de tous les programmes de calcul possibles. Donc, en connaissant Ω , on pourrait résoudre le problème de l'Arrêt (voir le chapitre sur les machines de Turing et la computation). Le problème est que les premiers n digits d' Ω ne peuvent pas se calculer en se servant d'un programme significativement plus court que n bits de longueur, ce qui est aussi le fondement du théorème d'incomplétude

de Chaitin utilisant l'AIT, basé sur l'idée qu'un ensemble d'axiomes ne peut pas prouver un théorème contenant plus d'information que la combinaison des axiomes eux-mêmes. Autrement dit, Ω est en lui-même non-calculable, car un programme d'une longueur fixe ne serait capable que d'estimer un petit nombre fini des digits du nombre infini Ω .

Bien qu' Ω possède une définition mathématique simple, la définition ne nous rend pas capables de calculer plus qu'un nombre fini de ses digits, et des nombres Ω ont été construits de façon à ce qu'aucun bit ne puisse être déterminé même avec la pleine puissance de la Théorie des Ensembles, qui est la théorie classique la plus puissante communément utilisée en mathématiques. Cependant, on sait qu'il existe des nombres Ω plus faciles que d'autres (Solovay [Solovay,]), On a vu qu'il peut y avoir un Ω pour tous les langages de programmation (autodélimités) et pour toutes les machines de Turing universelles (à préfixe libre) qui changent seulement les digits et l'ordre des bits dans l'expansion d' Ω . Cela pourrait se voir comme une question de reformulation en un autre langage de programmation. Une chose intéressante est qu'on sait aussi qu'il y a un nombre Ω (ou une famille de ces nombres) qui sont compliqués de façon maximum, au sens où aucun de ses digits ne peut se calculer, pas même un seul. Donc même quand on peut reformuler les questions sous la forme de programmes informatiques, afin d'essayer de tirer quelque information d' Ω , la plupart des questions ne vont pas seulement rester sans réponses, mais dans certains cas (pour une question malchanceuse, en quelque sorte), pas une seule réponse ne va pouvoir être extraite de certains nombres Ω .

La théorie algorithmique de l'information peut ainsi faire une surprenante contribution au débat philosophique sur les origines et les limites de la connaissance. Selon Gregory Chaitin, l'AIT révèle que certains faits mathématiques sont vrais sans raison de l'être, une découverte qui va à l'encontre du principe de Leibniz de raison suffisante, mais qui suit la dichotomie de Thomas d'Aquin sur le savoir connaissable et inconnaissable, qui sont d'une part la connaissance ou les vérités qui peuvent être connues et possèdent une explication, et d'autre part la connaissance ou les vérités qui le sont pour des raisons qui ne peuvent être connues ou atteintes par aucun moyen objectif.

1.12.4 Principe heuristique de Chaitin

Chaitin propose une manière alternative pour démontrer les limites de la méthode déductive axiomatique, le fait qu'une théorie formelle déductive ne peut pas être à la fois consistante et complète [Gödel, 1986], on l'appelle le

principe heuristique de Chaitin.

(principe heuristique de Chaitin [[Chaitin, 1975](#)]) : Un système axiomatique T ne permet pas de prouver une formule t si $K(T) \leq K(t) + c$ ou plus précisément, T ne peut pas produire un théorème t tel que t détermine que la complexité d'un objet quelconque est plus grand que $K(T)$.

L' Ω_T de Chaitin est en réalité une famille de nombres car il dépend du choix de la machine universelle (sans préfixe) de Turing T pour calculer les chiffres d' Ω . On ne peut pas connaître ces suites aléatoires comme on ne peut pas résoudre le problème de l'arrêt [[Turing, 1936](#)] des machines de Turing, comme on ne peut pas non plus décider si un théorème appartient, en général, à une théorie axiomatique quelconque [[Gödel, 1986](#)].

S'interroger sur les chiffres d'une suite aléatoire de cette famille est alors équivalent à s'interroger sur l'arrêt d'une machine de Turing (dont le meilleur exemple et précisément celle du nombre de Chaitin).

Selon Chaitin, et d'autres philosophes de la science, il y a d'autres propriétés intéressantes d' Ω , la plus intéressante est peut-être le propos sur l'aléatoire dans les mathématiques, car les chiffres d' Ω , et donc certains machines de Turing qui encodent des questions calculables (ou computables), comme par exemple des questions sur l'arithmétique, suggèrent qu'il y a une sorte d'aléatoire même à l'intérieur des mathématiques, car il y a des énoncés mathématiques qui sont vrais ou faux, d'après Chaitin, sans aucune raison et dans ce sens Ω tel que concept fondamental ajoute à ce que Gödel et Turing ont trouvé au sein des mathématiques avec leurs théorèmes d'incomplétude et d'indécidabilité.

En effet, comme il se trouve, d'après le nombre Ω , un nombre infini d'énoncés mathématiques sont irréductibles, au sens où aucune théorie effective (computation) ne peut les expliquer, ce qui veut dire que le seul moyen de prouver de tels énoncés est de les supposer axiomes, qui n'ont besoin d'aucune explication et sont supposés être vrais par définition. Mais même en supposant de nouveaux axiomes, il reste toujours un nombre infini de faits qui ne peuvent pas être prouvés vrais ou faux. C'est en ce sens que Chaitin prétend que, contrairement à la croyance courante qui veut que les mathématiques soient étrangères à l'aléatoire, les mathématiques sont aléatoires au sens profond du terme d'après Chaitin, car elles contiennent des vérités qui sont vrais pour aucune raison. Pour la relation précise entre nombre Ω et Problème de l'arrêt voir [[Chaitin, 2001](#), [Chaitin, 2008](#)]. Chaitin lui-même a exploré aussi les angles philosophiques d' Ω dans plusieurs ouvrages [[Chaitin, 2001](#), [Chaitin, 1975](#), [Chaitin, 2006](#), [Chaitin, 2008](#)].

Parmi les propriétés connues du nombre Ω de Chaitin, celles qui nous intéressent dans ce travail sont :

- Le nombre Ω est non calculable, comme la complexité algorithmique, car aucun algorithme ne peut égrener les chiffres de Ω un par un (toujours à cause du problème de l'arrêt).
- Le nombre Ω est aléatoire dans le sens suivant : le plus court programme qui engendre ses n premiers bits possède environ n bits. Aucun des nombres transcendants classiques ne possède cette propriété. D'ailleurs, les constantes π et e sont facilement compressibles puisqu'on connaît des algorithmes qui en calculent tous les chiffres un par un.
- La connaissance de m bits de Ω permet de savoir pour tout programme p dont la longueur est inférieure à m , s'il s'arrête ou non.

1.13 La profondeur logique de Bennett

Une autre tentative algorithmique fondée sur des notions de la théorie de la complexité algorithmique, est celle proposée par Charles Bennett. Complexe peut signifier aléatoire ou fortement structuré, riche en information. La complexité algorithmique, par exemple, classera les objets suivants de moins à plus complexes :

1. un cristal, objet répétitif.
2. un ordinateur, un être vivant.
3. un gaz, un nuage.

Pourtant, c'est sur le type 2 qu'on trouve les objets les plus organisés. C'est la critique formulée par Ilya Prigogine [Nicolis and Prigogine, 1989] au sujet de la complexité algorithmique comme mesure inadéquate de richesse en organisation.

Une tentative pour définir cette *complexité organisée* est la “profondeur logique” de Charles Bennett [Bennett, 1988].

Définition 5 [Bennett, 1988] : La profondeur logique d'une suite s est définie par :

$$P(s) = \text{temps de calcul du programme minimal de } s.$$

Ce qui tient compte à la fois de la complexité algorithmique (la taille du programme qui engendre s) et du nombre de pas du calcul que ce programme

prend pour engendrer s . La profondeur logique tente de mesurer le “contenu en calcul” d’un objet fini.

Une série d’arguments variés montre qu’une forte profondeur logique est la marque attestant qu’un objet est le résultat d’un long processus d’élaboration. Le temps de fonctionnement du programme minimal qui correspond à la complexité organisée d’une suite s’oppose à la complexité aléatoire définie par la complexité algorithmique.

La profondeur logique de Bennett est définie par le temps d’exécution requis pour générer une chaîne à l’aide d’un programme proche de l’incompressible, c-à-d qu’on ne peut pas le produire grâce à un programme significativement plus court. Logiquement, les objets profonds contiennent des indices internes d’avoir été le résultat d’une longue computation, et satisfont à une loi de croissance lente (par définition). On parlera davantage de la profondeur de Bennett dans les chapitres suivants.

1.14 La contingence de l’aléatoire quantique

“Quantum mechanics is the origin of randomness”.

Charles Bennett [[Bennett, 2008](#)]

(Une croyance partagée aussi par Seth Lloyd [[Lloyd, 2006](#)])

C’est l’inégalité de Bell et la nature non-locale de la mécanique quantique qui suggèrent que cette dernière est capable de produire de l’aléatoire d’une nature non-déterministe (car la violation du théorème de Bell signifie qu’il est impossible de construire une théorie locale déterministe à variables cachées rendant compte des résultats expérimentaux et donc que les mesures au niveau quantique sont réellement probabilistes et les états intrinsèques correctement décrits par la mécanique quantique).

Il faut aborder ces questions car le sujet reste à l’état de problème ouvert (la violation du théorème de Bell a d’autres solutions possibles, mais également ou encore plus difficiles à accepter car elles contredisent par exemple la théorie de la relativité).

C’est une question ouverte parce qu’il faut supposer que la théorie quantique est correcte et qu’effectivement ces propriétés étranges sont présentes pour arriver à la conclusion de la nature non-déterministe de l’aléatoire quantique.

D’un autre côté, il reste à savoir si on peut extraire de la valeur additionnelle de l’accès aux suites aléatoires non-déterministes. Si c’est le cas, du fait

qu’une machine universelle de Turing est incapable de produire de l’aléatoire non-déterministe, les générateurs des nombres aléatoires quantiques sont les appareils les plus proches de ce qu’on appelle un oracle dans la théorie de la calculabilité [Calude and Casti, 2004, Calude et al., 2010, Calude, 2005], mais il se pourrait que même l’accès à un tel pouvoir reste limité par ce qu’on peut interpréter ou manipuler (qui semble toujours être d’une nature finie et limitée).

La valeur pour la cryptographie cependant semble évidente, mais il se pourrait, comme plusieurs auteurs le soupçonnent, qu’aucun vrai avantage n’existe en pratique face aux bons générateurs de nombres aléatoires déterministes.

1.15 La structure cachée des nombres

En général, étant donné une suite quelconque il n’est pas facile de la distinguer comme aléatoire, même avec un seuil justifié de certaine manière. Par exemple, cette suite du semble aléatoire au sens de la complexité algorithmique :

26535897932384626433832795028841971693993751058209

mais elle ne l’est pas car il s’agit en fait de la suite des décimales du nombre π à partir du sixième chiffre (les cinq premières sont 14159). π n’est donc absolument pas algorithmiquement aléatoire, puisque le simple fait que l’on puisse l’écrire sous formes de suites permet d’écrire un programme de taille bien inférieure à n pour calculer n décimales.

On doit donc considérer les décimales du nombre π comme un exemple de complexité organisée⁹ (ou de complexité aléatoire faible) car π peut être produit par des programmes courts qui engendrent des segments initiaux de chiffres de π .

L’idée c’est que l’organisation de π est cachée, mais elle est extrêmement ordonnée et donc algorithmiquement non-aléatoire.

La suite ci-dessous semble à première vue aussi aléatoire :

9. complexe peut aussi signifier : organisé, fortement structuré, riche en information. Il y a deux types de complexité : la complexité aléatoire et la complexité organisée. On parlera ici plutôt de la complexité aléatoire.

0110100110010110100101100110100110010110011010010110100110010110

Cependant, elle est construite d'une façon récursive, par la concaténation des suites précédentes niées :

0

01

0110

01101001

0110100110010110

...

$tm(n+1) = tm(n) \text{ <> } \neg tm(n)$

Elle vient donc d'un algorithme assez simple (voir court). L'idée c'est qu'en général, la complexité aléatoire n'est pas facile à distinguer.

1.15.1 Un îlot dans l'océan de l'irréductibilité Raccourcis de π et l'irréductibilité de Wolfram

Les décimales de π n'ont rien d'aléatoire selon la théorie algorithmique de l'information, mais la distribution des chiffres ou des groupes consécutifs de n chiffres de ses décimales a cependant les caractéristiques d'un phénomène aléatoire.

Les formules BBP (Bailey-Borwein-Plouffe) [Bailey et al., 1997] permettent d'évaluer le n -ième chiffre de π en base 16^{10} sans calculer les précédents, le tout en temps $n \times \log(n)$. Il est pertinent de remarquer que la complexité algorithmique ne prend pas en compte le temps du calcul (le nombre de pas qu'un calcul exige).

D'après Stephen Wolfram [Wolfram, 2002], les raccourcis ne existent pas en général (par exemple, pour la règle 30 des automates cellulaires). D'après lui cela provient du fait que tout système de calcul non-trivial atteint un pouvoir de calcul également sophistiqué. Un système ne peut pas raccourcir le calcul d'un autre sans passer par toutes les étapes (ou la plupart des étapes) précédentes du système qu'on essaye de raccourcir [Dubucs, 2006].

10. ou en n'importe quelle base multiple de 2^n .

Soit parce qu'on ne peut pas trouver le raccourci soit parce qu'il n'existe pas (Wolfram ne prend pas de position philosophique à cet égard). Wolfram appelle ce phénomène *principe d'irréductibilité du calcul*.

Une tentative pour expliquer ce phénomène, c'est d'utiliser la "profondeur logique" P de Charles Bennett. La profondeur logique exprime le "contenu en calcul" d'un objet fini [Bennett, 1988]. Rappelons que $P(s)$ = temps de calcul du programme minimal de s . Ce qui tient compte à la fois de la complexité algorithmique (la taille du programme qui engendre s) et du nombre de pas du calcul que ce programme prend pour l'engendrer. Donc π ne sera pas complexe ni au sens de la complexité algorithmique de Kolmogorov ni au sens de la profondeur logique de Bennett, car il est compressible et facilement calculable. Par contre, il reste à savoir si le raccourci trouvé pour π est une exception à la règle ou si la difficulté des raccourcis est non leur existence mais la difficulté de les trouver pour une infinité (non dénombrable) de cas.

1.15.2 Principes d'irréductibilité

Une conséquence du principe d'irréductibilité de Wolfram, c'est que des systèmes simples qui atteignent un pouvoir de calcul minimal peuvent produire une sortie si complexe qu'aucun autre système ne peut la déchiffrer par avance (le raccourcir) sans parcourir le calcul étape par étape, ce qui donne lieu à la définition de l'aléatoire déterministe de Wolfram.

Pour illustrer ce phénomène Wolfram utilise des exemples d'automates cellulaires (voir figs 1.15.2 et fig 1.8).

Un automate cellulaire est un modèle de calcul. Il consiste en une grille régulière de "cellules" contenant chacune un symbole choisi parmi un ensemble fini. Le symbole d'une cellule au temps $t + 1$ est fonction du symbole au temps t dans la même cellule et/ou dans le voisinage, c'est-à-dire, les cellules à certain distance de la cellule en temps t . À chaque nouvelle unité de temps, les mêmes règles sont appliquées simultanément à toutes les cellules de la grille en appliquant une règle locale qui prenne en considération les symboles du voisinage. L'ensemble des règles locales engendrent une règle globale. Pour des automates cellulaires linéaires avec 2 symboles (donc binaires) et voisinage 2 (donc 3 cellules; une de chaque côté), Wolfram a proposé une énumération qui consiste à appeler ces règles par leur représentation binaire. Ainsi, la règle 1 par exemple (un automate dite élémentaire), est la règle qui assigne à chaque cas possible parmi les 2^3 (111, 110, 101, 100, 011, 010, 001, 000) les valeurs 0,0,0,0,0,0,0,1, c'est-à-dire, pour tous les cas sauf pour 000

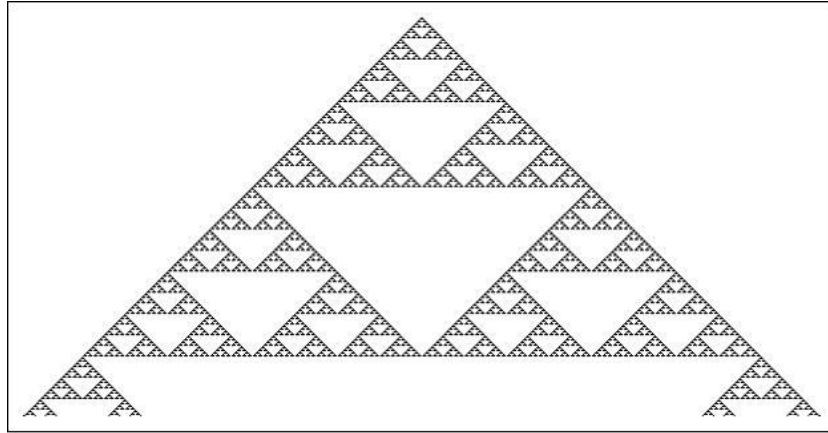


FIGURE 1.4 – Automate cellulaire élémentaire à règle 22. Le temps parcourt en vertical. La première rangée est la condition initiale (une cellule noire). À chaque lignée on applique la règle 22 pour produire la suivante.

on imprime le symbole 1, autrement on imprime toujours 0. Cela signifie que si par exemple, à un temps t donné, une cellule est “0”, sa voisine de gauche “0” et sa voisine de droite “0”, au temps $t + 1$ la cellule centrale sera “1”, pour tout autre cas elle sera “0”. Voir les figures ou pour des autres exemples concrets de règles d’automates cellulaires élémentaires.

L’universalité de Turing existe chez les automates cellulaires. On peut définir cette notion comme la capacité d’un automate à simuler une machine de Turing universelle. En effet, par exemple l’automate élémentaire à règle 110 avec une condition initiale particulière, a été prouvé être capable de simuler tout fonction calculable par une machine de Turing. En effet Cook et Wolfram ont montré que dans ces conditions la règle 110 peut émuler n’importe quel système de *tag cyclique* (qui sont universels) et par conséquent n’importe quelle machine de Turing [Cook, 2004, Wolfram, 2002].

Dans un automate cellulaire à aléatoire extrinsèque, l’automate laisse passer (amplifie peut-être) l’aléatoire vers la sortie mais la source de l’aléatoire vient de l’extérieur (par exemple il pourrait s’agir de la règle 22). Par contre, un automate à règle 30 n’exige pas d’avoir comme entrée une suite aléatoire pour produire comme sortie une suite qui semble aléatoire (au moins pour la partie droit).

Ces *principes d’irréductibilité* sont complémentaires des résultats d’indécidabilité (et en particulier celle du problème de l’arrêt). Une procédure de décision est bien une sorte de raccourci, et donc le problème de l’arrêt est

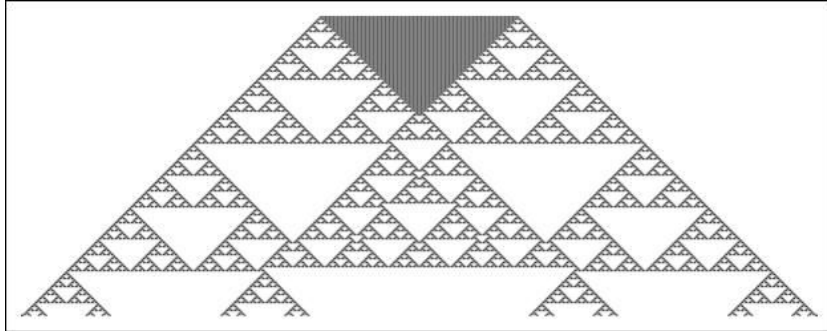


FIGURE 1.5 – Un automate qui préserve la nature des entrées : À des entrées simples, des sorties simples. À des entrées complexes des sorties complexes : Automate cellulaire élémentaire à règle 22 : condition initiale périodique $n = 2$ (...010101...).

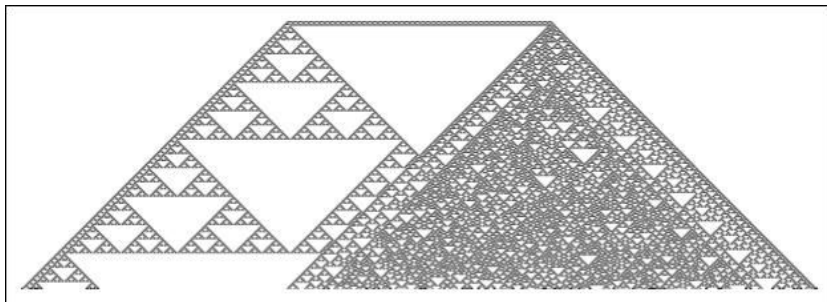


FIGURE 1.6 – Brisure de symétrie : Automate cellulaire élémentaire à règle 22 : condition initiale périodique asymétrique $n = 3$ (...01101...). Ce type d'automates réagissent à la symétrie ou asymétrie de la condition initiale.

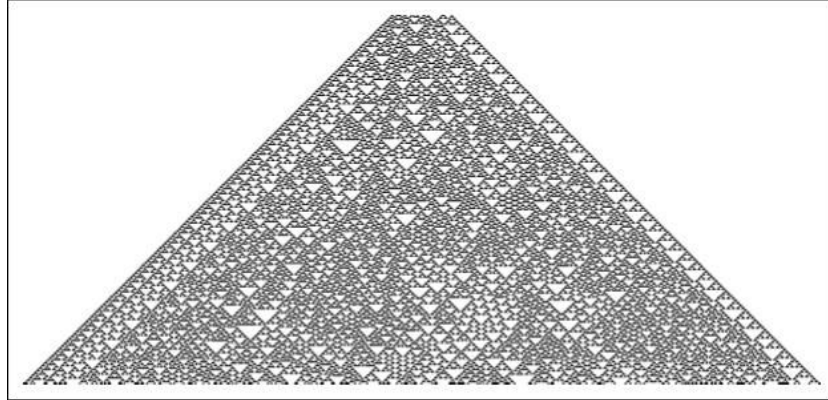


FIGURE 1.7 – L'aléatoire extrinsèque venu d'un système déterministe : Automate cellulaire à règle 22 à condition initiale aléatoire.

l'un des principes d'irréductibilité les plus forts (ainsi que d'autres problèmes équivalentes, comme le problème de l'accessibilité ou *reachability* en anglais). Un autres principe d'irreducibilité est ceux du problème de calcul de temps représenté par la question **P** versus **NP** (**P** pour temps Polynomial et **NP** pour temps polynomial avec un machine de Turing non déterministe). On croit que certains problèmes qui peuvent être résolus en temps polynomial par des machines non déterministes ne peuvent pas l'être en temps polynomial par des machines déterministes, ou autrement dit, des problèmes dont il n'y a que de solutions qui prennent du temps exponentiel avec une machine déterministe. Par contre, ces questions de temps de calcul exigent un définition asymptotique en fonction d'une entrée de taille croissant ce qui n'est pas nécessairement le cas pour les questions de raccourcis (par exemple, pour le temps pour calculer n'importe quel nombre des chiffres de π). De plus, ces concepts d'irréductibilité sont différents aussi du problème des systèmes dynamiques chaotiques dont on exige un concept de mesure et de conditions initiales arbitrairement proches qui ne sont pas non plus présentes dans les problèmes des raccourcis pour, par exemple, un automate cellulaire. Une tentative de encadrer ce concept de raccourci à la Wolfram dans la théorie de temps de calcul (auquel appartient le probleme sur **P** et **NP**) est apporté par Zwirn et Delahaye in [Zwirn, 2013] and [Zwirn, 2015].

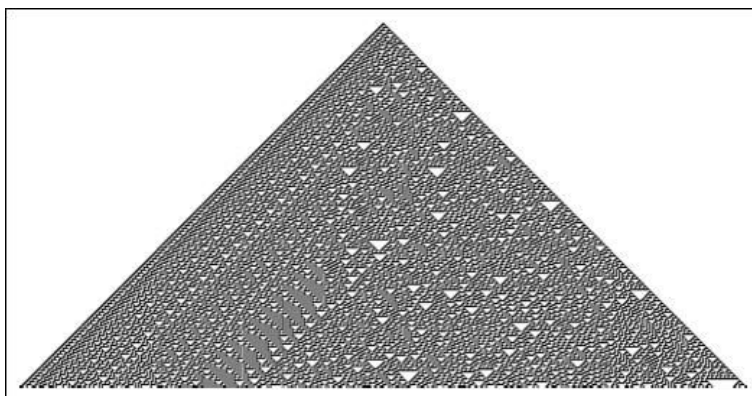


FIGURE 1.8 – Automate cellulaire à règle 30.

1.15.3 L'aléatoire intrinsèque

D'après Wolfram, la sortie de l'automate cellulaire à règle 30 produit une sortie assez complexe pour être considérée comme aléatoire. Il appelle ce type d'aléatoire venu du système même (opposé à l'aléatoire venu de l'extérieur et transporté/transféré vers la sortie) de *l'aléatoire intrinsèque*.

La fig. 1.8 montre un système qui produit de l'aléatoire même à partir de conditions initiales simples (voire même d'une cellule noire).

Pour Wolfram, ce type d'aléatoire comme celui de la règle 30¹¹ est aussi bon en qualité que celui venu des systèmes non-déterministes (s'il y en a, car Wolfram ne se prononce pas à cet égard).

Pourtant, l'aléatoire algorithmique s'oppose évidemment à l'aléatoire intrinsèque de Wolfram car ce dernier, comme pour la règle 30, vient d'une règle très courte qui génère l'évolution complète de l'automate (on n'a besoin que de 8 bits pour le décrire).

Cependant, lorsqu'on essaie de compresser la sortie de l'automate à règle 30 aucun compresseur ne semble bien la compresser, comme cela arrive aussi pour les chiffres décimaux du nombre π . Aucun compresseur ni l'analyse soigneuse des chercheurs n'ont réussi à trouver des motifs réguliers dans la partie droite de la règle (par exemple, une période).

11. Wolfram utilise un automate cellulaire comme générateur de nombres aléatoires dans le logiciel Wolfram *Mathematica* (et le langage de programmation Wolfram Language), ce qui rend évident la valeur qu'il donne à ses croyances sur la qualité de l'aléatoire de systèmes déterministes tel que les automates cellulaires.

Dans le chapitre suivant je présente les résultats d'un test qui pourrait encourager la mise en œuvre d'approches expérimentales pour cerner la notion d'irréductibilité computationnelle. Ce test implique des tentatives systématiques pour court-circuiter la computation d'un grand nombre de petites machines de Turing (toutes à 3 et 4 états, 2 symboles) grâce à la prévision de suites d'entiers, en utilisant un programme spécialisé dans la recherche de fonctions. Cette expérimentation massive encourage une investigation sur les taux de convergence des procédures de décision, sur la décidabilité des ensembles, tout en incitant à discuter sur l'(im)prévisibilité des systèmes déterministes de calcul, en pratique. Je pense que cette investigation constitue une nouvelle approche quant au débat sur la question épistémologique dans le contexte de la simulation informatique, et représente ainsi une intéressante exploration à la limite entre les préoccupations philosophiques et les expériences computationnelles.

1.16 Résumé de la première partie

Dans la Théorie Algorithmique de l'Information on peut distinguer deux grands sous-domaines. L'un concerne l'étude de la complexité algorithmique des chaînes finies, l'autre concerne l'étude de l'aléatoire algorithmique des chaînes infinies (ou des ensembles ou des nombres réels). Les deux sont profondément reliés, et portent en eux des contenus riches en débats, mais dans le même temps, ces deux sous-domaines affichent d'importants désaccords et incompatibilités, qui sont fascinants à étudier et à essayer de comprendre. Est-ce causé par une certaine différence essentielle ? Ou le désaccord est-il seulement passager ?

La convergence des définitions dans la Théorie Algorithmique de l'Information a rassemblé des caractérisations de l'aléatoire qu'on était loin de voir comme aussi profondément connectées formellement, telles que l'imprévisibilité, la compressibilité et la régularité. Dans ce bref compte-rendu de l'AIT, on a expliqué que la Théorie Algorithmique de l'Information était une théorie puissante fournissant un ensemble de mesures universelles (qu'on peut appliquer de la même façon à n'importe quelles données et la mesure est en fait la même à la limite) de la complexité pour des données concernant des objets individuels, notamment des chaînes et des séquences. Une propriété courante de ces mesures selon l'AIT est la forme variée de leur incalculabilité, dont nous avons expliqué qu'il ne faut pas le prendre comme un empêchement à leur utilisation, ni théoriquement ni expérimentalement, pas seulement parce que certains cas simples sont calculables, mais surtout parce que ces mesures

non complètement calculables sont approchables avec des outils tels que les algorithmes de compression, qui sont des preuves suffisantes du caractère non-aléatoire d'un objet.

Nous avons aussi établi que le traitement quantitatif par l'AIT, au sujet de l'information et du contenu informatif (sous la forme de l'aléatoire), ne devait pas être pris comme une disqualification de cette théorie quant à l'obtention d'important éclairages sur des questions philosophiques profondes touchant à l'information, spécifiquement sur la sémantique de l'information. Nous avons vu aussi que cette (demi)calculabilité est aussi l'atout principal de la théorie, qu'aucune mesure (totalement) calculable ne peut cerner, et que l'approche par la probabilité algorithmique est profondément liée à d'anciennes questions ouvertes sur l'inférence inductive, l'épistémologie et même la psychologie.

On peut se demander si l'aléatoire égale la complexité. L'idée intuitive de complexité suggère habituellement l'ordre et la structure, quelque chose qui ne conduit pas au concept de complexité de Kolmogorov signifiant l'aléatoire. Mais il y a une autre importante mesure de la complexité qui, à partir du concept de complexité de Kolmogorov, amène jusqu'à saisir l'idée intuitive de complexité structurelle, dont nous avons l'intention de parler dans des versions futures. Pour le moment, il s'agit d'avoir une meilleure vision de ce que les mathématiques peuvent faire pour quantifier les notions de simplicité et les niveaux d'aléatoire.

- La définition de l'aléatoire algorithmique pour des suites finies ne distingue pas les suites aléatoires de celles qui ne le sont pas, mais elle distingue ce qui ne semble pas aléatoire de ce qui le semble par rapport à la taille de la suite originale.
- Pour la définition de l'aléatoire pour des suites finies, la taille de la suite joue un rôle important, tandis que pour la définition de l'aléatoire pour des suites infinies, la question ne se pose pas.
- La définition de l'aléatoire pour des suites infinies est ontologiquement robuste mais épistémologiquement faible.
- La définition de l'aléatoire pour des suites finies est épistémologiquement plus forte, mais relative et subjective.
- Bien que la définition de l'aléatoire pour des suites infinies soit épistémologiquement faible, elle est méta-épistémologiquement riche.
- Il semble y avoir une distribution qui donne de la stabilité à la définition de la complexité algorithmique et donc la rend indépendante du langage. Il semble possible alors de se débarrasser du problème du choix du langage et donc des choix arbitraires pour définir et mesurer la complexité algorithmique.

- La théorie de la complexité algorithmique n'adopte aucune présomption à l'égard de la nature déterministe ou indéterministe du processus générateur. Cependant la distribution obtenue des événements détermine, avec probabilité $m(s)$, si à l'origine des données se trouve un processus de calcul.
- Finalement, les approches algorithmiques capturent bien le phénomène en correspondance avec les notions intuitives, y compris la difficulté de calculer et d'exhiber ce qu'est l'aléatoire.
- La vraie nature de l'aléatoire quantique et sa caractérisation algorithmique est une question qui reste ouverte (même si on y fait des progrès [[Calude and Svozil, 2008](#)]).
- Une caractérisation par le moyen du concept d'irréductibilité est intéressante et proche de celle de l'imprédictibilité.

Deuxième partie

De l'incalculable à l'irréductible

Chapitre 2

Rencontres Empiriques avec l'Irréductibilité et l'Imprévisibilité dans le Calcul

2.1 Une approche informatique expérimentale d'un débat épistémologique

Il y a des siècles que l'incertitude formelle intéresse les penseurs. Le paradigme mécaniste, consolidé par les travaux de Boyle, Hooke, Newton et Laplace, entre autres, a dominé la science moderne en remplaçant les théories téléologiques antérieures. Toutefois Popper [Popper, 1988], en contestant le déterminisme scientifique, a fait une distinction empirique entre les horloges et les nuages quant à leur (im)prévisibilité.

Avec le développement de la mécanique quantique et la découverte du phénomène de chaos déterministe, l'approche mécaniste, qui insistait sur la vision d'un monde horloger à la prédictibilité sans limites, commença à paraître fragile. Tandis que la mécanique quantique appuyait son imprévisibilité sur l'incertitude régnant au niveau des propriétés intrinsèques des particules [Blum, 1971], l'étude des phénomènes non-linéaires a opéré une déconnexion entre déterminisme et prédictibilité. À l'aide de simulations informatiques, Edward Lorenz [Lorenz, 1963] découvrit que certains systèmes physiques étaient instables face à de petites modifications. En gros, après un court moment, leur comportement devient imprévisible, lors de petits changements dans leurs conditions initiales, même si les systèmes sont pleinement déterministes.

On a présenté divers arguments et résultats au sujet de toutes sortes de limites épistémologiques associées à toutes sortes de formes d'irréductibilité, dans le calcul digital, et ceci indépendamment des conditions physiques (c'est-à-dire surtout indépendamment des ressources dont un système peut avoir besoin pour mener un calcul). Ces résultats sur l'irréductibilité sont inhérents aux systèmes de calcul, étant incarnés par l'indécidabilité du problème de l'arrêt. Le théorème de Turing sur le problème de l'arrêt [Turing, 1936] établit qu'il n'existe aucun algorithme pour déterminer si une machine de Turing, avec une entrée donnée, va s'arrêter ou va continuer à tourner indéfiniment. Et le problème de l'arrêt résiste en calcul digital, quel que soit la quantité des ressources physiques disponibles. Nous nous intéressons aux mesures de l'imprévisibilité qui, comme pour le problème de l'arrêt, sont indépendantes des ressources physiques et ne peuvent venir que des propriétés intrinsèques du calcul digital.

Parmi d'autres types d'irréductibilité, il existe un phénomène interne au déploiement évolutif des systèmes déterministes, qui appartient à un type non décrit par les études précédentes sur les mesures d'irréductibilité, c'est le type suggéré par le principe de Wolfram sur l'irréductibilité. Ce principe affirme que même si des systèmes triviaux peuvent permettre la réduction du nombre d'étapes d'un calcul en utilisant un autre système, il existe des computations non-triviales qui ne tolèrent aucun raccourci, sauf à parcourir (presque) toutes les étapes plus rapidement. Comme pour tous les principes ou lignes directrices, une partie de ce principe reste intuitive. Nous pensons que l'étude de ce genre de questions épistémologiques a commencé à changer avec l'utilisation des ordinateurs et des opportunités expérimentales qu'ils offrent, comme Wolfram l'a souligné lui-même [Wolfram, 2002].

Depuis plus d'un demi-siècle, mathématiciens et informaticiens savent que la victoire finale n'aura pas lieu dans la bataille pour atteindre la prédictibilité ultime. Même pour les systèmes déterministes, on a accepté l'existence de vérités et de faits, au sujet de ces systèmes, qui vont rester non-prouvés, comme Kurt Gödel et Alan Turing l'ont montré pour les systèmes axiomatiques formels et les machines calculantes. Cela ne signifie pas qu'on doive renoncer à récolter de nouveaux indices à travers l'étude de la prévisibilité et de l'imprévisibilité dans les systèmes déterministes. Je propose ici une approche de "philosophie expérimentale" au problème.

Des machines calculantes de plus en plus puissantes rendent possibles des tâches qui jusqu'ici semblaient impossibles, et il n'a pas fallu longtemps après les travaux de Gödel et Turing pour qu'on commence à se demander combien de temps une tâche pourrait prendre si elle était réalisée avec tel ou tel algorithme. Le besoin pour une notion de complexité devint clair. Bien que

non clairement énoncé, dans une lettre écrite à von Neumann (Princeton, 20 Mars 1956), Gödel lui-même avait déjà proposé une première version du problème **P** versus **NP**?, en termes de problèmes finis de durée quadratique (cette lettre vint au jour en 1988). L'efficacité devint un principe directeur dans la conception des algorithmes, et on découvrit que l'idée de concevoir un algorithme capable de produire des algorithmes optimaux était équivalent au problème de l'arrêt. Car s'il y avait moyen de trouver un tel algorithme, les problèmes d'insolubilité et d'indécidabilité de Gödel et Turing seraient résolus. Donc on n'a pas d'autre choix que d'échanger nos objectifs idéalistes pour des approximations réalistes. Dans ce papier, nous présentons une approche expérimentale pour traiter de la notion d'irréductibilité computationnelle, en tant que reliée au problème d'imprévisibilité empirique. Nous pensons que notre approche est novatrice, car nous donnons un cadre formel (une expérimentation informatique) à une notion que nous croyons être largement épistémologique par nature (l'irréductibilité computationnelle).

2.2 Prévoir le comportement de systèmes de calcul déterministes

Observer le comportement d'un système pendant un certain temps et dans certains cas va-t-il nous apprendre quelque chose sur le comportement du système dans le futur et dans un cas plus général? Par exemple, la mesure de transition de phase présentée en [Zenil, 2010] implique qu'on puisse, dans certaines limites, être capable de prédire le comportement global d'un système à partir d'un segment d'entrées initiales, ceci basé sur la variabilité antérieure du dit système. L'expérience nous dit qu'on pourrait bien prévoir le comportement futur en se basant sur le comportement antérieur (une hypothèse bayésienne), pourtant nous savons que c'est impossible dans le cas général à cause du problème de l'arrêt.

C'est lié aux découvertes d'Israeli et de Goldenfeld [Israeli and Goldenfeld, 2004]. Ils ont montré que certains Automates Cellulaires Élémentaires (ECA) computationnellement irréductibles possèdent des propriétés prévisibles dans certains cas de réduction à gros grains. Ils ont procédé en utilisant une technique de renormalisation de groupe, un dispositif mathématique qui permet de rechercher les modifications dans un système physique, en l'observant à différentes échelles de distance. Ils ont cherché les moyens de remplacer plusieurs cellules d'un automate par une cellule unique. Toutefois, ces capacités de prédiction sont plombées par l'inévitable (et finalement indécidable) problème de l'induction, en l'occurrence

de savoir si on peut généraliser la prévision pour toutes conditions initiales et tout nombre d'étapes, sans avoir à dérouler le système pour toutes les conditions initiales possibles et pour un nombre arbitraire d'étapes.

La question est donc de savoir, dans certaines circonstances, si ce type de prévision à grande échelle est possible. Par exemple pour l'ECA de règle 30, selon la nomenclature par énumération des CA d'après Wolfram [[Wolfram, 2002](#)], cette approche à grande échelle ne semble pas vouloir dire grand chose. Dans de nombreux cas, il semble tout au plus qu'on puisse prédire quelques étapes futures, ceci voulant dire que même si le comportement du système n'est pas complètement chaotique, il est imprévisible, au sens où on ne peut généralement pas prévoir un nombre arbitraires d'étapes sans avoir à dérouler la computation totale étape par étape. Pour de nombreux systèmes, dont les plus apparemment aléatoires, on ne peut pas s'attendre à un comportement général très différent. Une des caractéristiques essentielles des systèmes appartenant à la classe 4 de Wolfram [[Wolfram, 1984](#)] est précisément l'existence de structures envahissantes qu'on ne peut prévoir que jusqu'à un certain point. Le surprenant dans ce phénomène, c'est que dans tous ces systèmes totalement déterministes et extrêmement simples, comme un ECA, on ne puisse pas trivialement prédire tous les aspects de leur évolution, et pas à cause d'un problème de mesure ou d'hypersensibilité, mais à cause du fait qu'on ne sait pas s'il existe des raccourcis – et s'il en existe, on ne sait pas comment les trouver systématiquement. Certaines de ces questions sont réductibles au problème de l'arrêt.

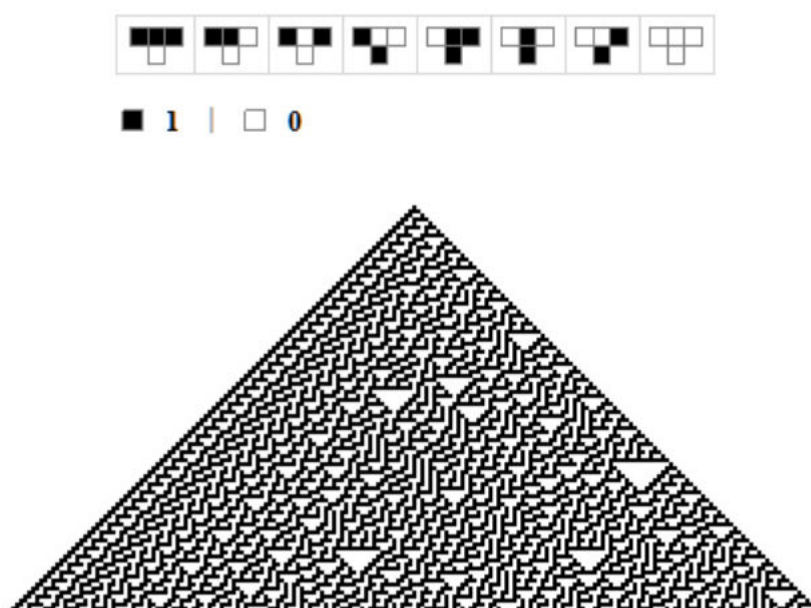


FIGURE 2.1 – Dans cette simulation de l'ECA de règle 30, chaque rangée de pixels est dérivée de celle qui est au-dessus. On ne connaît aucun raccourci pour calculer toutes les valeurs de bit de n'importe quelle colonne ou rangée, sans avoir à dérouler le système, malgré l'extrême simplicité de la règle pleinement déterministe qui le génère (les icones en haut de la figure).

2.3 Mesure de l'irréductibilité

Il y a plusieurs formes d'irréductibilité dans les systèmes de calcul, en partant de l'indécidabilité, en passant par l'intraçabilité et en allant jusqu'à la non-linéarité. Le principe d'irréductibilité computationnelle, comme mis en place dorénavant en [Wolfram, 2002], établit que même si certaines computations admettent des raccourcis permettant leur simulation plus rapide grâce à un autre système, la plupart des systèmes non-triviaux demanderaient une simulation dotée à peu près du même nombre d'étapes que les étapes du système simulé. Autrement dit, sauf à dérouler les étapes plus rapidement (disons avec un ordinateur plus rapide), les processus computationnels non-triviaux ne peuvent pas être simulés plus vite que la façon dont ils opèrent. Wolfram ne semble pas prendre position quant à savoir si c'est une question ontologique ou épistémologique. C'est à dire savoir si aucun raccourci n'existe, ou si on ne les connaît pas en général. Les auteurs pensent que le principe, dans le contexte originellement établi, admet une interprétation épistémologique.

Bien que l'irréductibilité computationnelle et la complexité computationnelle soient concernées par la durée, le principe de Wolfram est intrinsèque aux systèmes de calcul, au sens où il est indépendant de toute ressource extérieure, telles que les entrées fournies aux systèmes. La complexité temporelle traditionnelle est une mesure basée sur le comportement asymptotique d'un système par rapport à la longueur de ses entrées. On ne peut cependant pas dire si une machine avec une entrée fixe appartient à une classe particulière de complexité temporelle, alors que la question de l'irréductibilité computationnelle de Wolfram fonctionne toujours.

Nous n'allons donc pas discuter des mesures traditionnelles de complexité, car nous pensons qu'elles ne saisissent pas la notion d'irréductibilité computationnelle (et computationnelle, non pas, par exemple, chaotique), n'étant pas définies pour une computation partant d'une configuration initiale fixe donnée. D'autres mesures non-constructives de la complexité et leurs résultats associés, comme le théorème de l'accélération de Blum, bien que pertinent et en rapport, appartiennent à la branche hautement non-constructive de la théorie de la calculabilité. Pour certains, l'existence sans méthode constructive peut être aussi significative que la non-existence. Des théoriciens de la calculabilité se sont cependant arrangés pour présenter des résultats négatifs d'une façon positive, au moyen des encadrements hautement non-constructifs dans lesquels ils travaillent dorénavant.

Le théorème d'accélération de Blum affirme qu'il existe des problèmes

pour lesquels trouver l'algorithme optimal ne peut pas se faire. Pour tout algorithme qui résout un problème, il en existe toujours un autre qui soit significativement plus rapide. La démonstration du théorème de l'accélération est donnée en [Blum, 1971, Meyer and Fischer, 1972, Helm and Young, 1971, Schnorr, 1972]. Les arguments utilisés pour défendre la position disant que ce théorème n'est pas pertinent pour les calculs concrets, sont connus dans la littérature [van Emde Boas, 1975].

Des versions plus efficaces du théorème de Blum concernant l'accélération occasionnelle permettent à un algorithme de calculer une fonction plus rapidement pour un certain nombre de paramètres, et même pour un nombre infini d'entre eux, alors que le calcul avec le reste des paramètres peut être plus lent. Si on suppose que le théorème d'accélération de Blum est efficace pour un segment initial de valeurs d'une fonction, cela conduit à une étrange situation où on sait que le programme le plus rapide se trouve parmi un ensemble fini de programmes à peu près équivalents entre eux en efficacité, et qui calculent des fonctions dotées d'un nombre fini de valeurs incorrectes, introduisant par là un degré d'incertitude dans les capacités de prévision. Il semble donc qu'il y ait un rapport entre l'efficacité et la certitude de prévision.

Sans diminuer l'importance des approches théoriques comme celle du théorème d'accélération, on peut citer l'observation suivante attribuée à David Deutsch [Deutsch, 2011] :

La théorie de la computation a été étudiée presque uniquement dans l'abstrait, comme un sujet de mathématiques pures. On passe ainsi à côté de l'essentiel. Les ordinateurs sont des objets physiques, et les calculs sont des processus physiques. Ce qu'un calculateur peut ou ne peut pas calculer est déterminé par les seules lois de la physique, et pas par les mathématiques pures.

Sutner [Sutner, 2003] souligne aussi que Deutsch :

[F]ait une importante remarque : les types de computations qui peuvent être physiquement réalisés, au moins dans le contexte d'un certain modèle idéalisé de la physique, ne sont pas bien représentés par la théorie purement mathématique de la computation.

Notre approche est expérimentale, et constitue un test nouveau combinant une question philosophique avec une simulation informatique. Et nous ne sommes ni intéressés par l'incursion dans un débat ontologique sur les

relations entre mathématiques et physique, ni intéressés par les mesures purement théoriques de la complexité temporelle même si elles ont bien une certaine valeur pratique et prédictive. Nous nous intéressons plus aux conséquences de la conduite d'une expérience, c-à-d à la performance de cette simulation informatique basique comme un moyen de récolter des indices sur la notion d'irréductibilité computationnelle, et comme une contribution au débat sur cette question épistémique dans un contexte plus formel, bien qu'expérimental. Notre approche pourrait stimuler une recherche empirique plus avancée pour ce genre d'irréductibilité computationnelle.

Les tests consistent à faire en sorte qu'un algorithme essaye de prévoir un calcul, à partir d'un certain historique calculatoire donné. Grâce à la demi-mesure $m(s)$ de Levin [Levin, 1977], on sait qu'on peut construire une distribution a priori basée sur cet historique calculatoire, et qu'il n'y a pas meilleur moyen (en l'absence d'autre information) pour prédire un résultat que d'utiliser $m(s)$. Mais $m(s)$ n'est pas calculable. L'option alternative est donc d'appliquer une batterie d'algorithmes connus à une série de résultats pour voir si le résultat suivant peut être prédit. La chose évidente à faire est d'essayer de saisir le comportement de différents systèmes et voir si on peut dire quelque chose au sujet de leur évolution future. C'est l'approche que nous avons adoptée.

2.4 L'expérience

On va souvent se référer au groupe de machines de Turing à n états et 2 symboles, en tant qu'espace de machines de Turing noté $(n,2)$. On a fait tourner toutes les machines de Turing à ruban infini d'un seul côté dans $(2,2)$, et $(3,2)$ pendant 1000 étapes, pour les 21 premières valeurs d'entrée $0, 1, \dots, 20$. Si une machine de Turing ne s'arrête pas au bout de 1000 étapes, on dit qu'elle diverge (la computation n'atteint aucune valeur par un arrêt). Clairement, à la mise en place de ce projet nous avons eu besoin de prendre des décisions au moins sur les problèmes suivants :

1. Comment représenter les nombres sur une machine de Turing.
2. Comment décider quelle fonction va être calculée par telle machine de Turing.
3. Comment décider si une computation est considérée comme terminée. On a réuni toutes les fonctions pour les 21 entrées, et on a comparé les classes de complexité temporelle, le nombre de résultats et le nombre de différentes fonctions calculées entre $(2,2)$ et $(3,2)$, ainsi que

le nombre de résultats, pour déterminer quelle fonction était calculée par une machine de Turing.

2.5 Formalisme

Dans notre expérimentation, on a choisi de travailler avec des machines de Turing déterministes, à un seul côté et à un seul ruban, comme nous l'avons fait auparavant [Joosten et al., 2011] pour des expériences sur les compromis entre le temps et la complexité basée sur la taille des programmes. Ce qui revient à dire qu'on travaille sur des machines de Turing avec un ruban illimité à gauche, mais limité sur la droite. Chaque transition d'une étape dans le modèle classique des machines de Turing est définie en coût par une unité de temps. Les machines de Turing à un seul côté sont parmi les conventions les plus courantes dans la littérature, précédées probablement seulement par les machines à deux côtés. Les considérations suivantes nous ont conduit à préférer travailler avec les machines de Turing à un seul côté, que nous avons trouvées plus adaptées que d'autres configurations pour notre expérimentation.

Il existe $(2sk)^{sk}$ machines de Turing à ruban d'un seul côté et à s -états et k -symboles. Ce qui donne 4096 dans $(2,2)$ et 2 985 984 dans $(3,2)$. Le nombre de machines de Turing croît exponentiellement quand on ajoute des états. Pour représenter les données sans avoir à représenter la totalité de chaque résultat, ce qui aurait rapidement dépassé notre capacité hardware, il a fallu imaginer un schéma représentatif qui soit efficace au niveau de l'optimisation de l'espace (donc non-unitaire). Avec un ruban mono-côté illimité à gauche, et limité à droite, l'interprétation du contenu d'un ruban qui est presque uniformément zéro, est direct. Par exemple, le ruban ... 00101 serait interprété comme une chaîne binaire valant 5 en base 10. L'interprétation d'un digit dépend de sa position dans le nombre. Par ex. dans le nombre décimal 121, le 1 le plus à gauche correspond aux centaines, le 2, aux dizaines, et le 1 à droite aux unités. Pour un ruban infini à deux côtés, on peut imaginer de nombreuses façons d'obtenir une représentation, mais toutes semblent plutôt arbitraires.

Avec un ruban mono-côté, on n'a pas non plus besoin d'un état d'arrêt supplémentaire. On n'a qu'à dire qu'une computation s'arrête quand simplement la tête "s'effondre" sur le côté droit du ruban. Ce qui arrive quand la tête est sur la cellule la plus à droite, et reçoit l'instruction pour bouger à droite. Un ruban illimité à deux côtés demanderait un état d'arrêt supplémentaire, ce qu'on ne désire pas au vu de ces considérations. En explorant une classe finie entière, on évite de faire les choix d'une énumération, qui

sont toujours arbitraires ou difficiles à justifier. C’est pourquoi l’énumération employée dans notre approche exhaustive n’a pas de pertinence particulière, elle nous assure simplement qu’on inspecte chacune des machines dans l’espace donné, une et une seule fois. C’est bien sûr réalisable car chaque espace $(2,2)$ est fini.

Sur la base de ces considérations, et du fait que ce travail a suivi les grandes lignes des expérimentations réalisées en [Wolfram, 2002], nous avons décidé de fixer ce formalisme pour les machines de Turing, et avons choisi le modèle à ruban d’un seul côté. On a décidé de représenter l’entrée en unitaire. D’un point de vue théorique, il vaut mieux que l’entrée à ruban vide soit différente de l’entrée zéro. Donc le choix final de la représentation de notre entrée fut de représenter le nombre x par $x + 1$, et ainsi de suite en ajoutant des 1. Une discussion sur le choix de la représentation de l’entrée est disponible en [Joosten et al., 2011].

2.6 Définition d’une fonction calculée et choix de la durée avant coupure

En général, savoir si une fonction est définie par la computation d’une machine de Turing particulière, est une question indécidable, car il n’existe pas de procédure générale finie pour vérifier que $M(x) = M'(x)$ pour tout x . Savoir si deux machines de Turing définissent la même fonction est indécidable pour les mêmes raisons. Par convention, on dit alors qu’une machine de Turing M calcule une fonction f si $M(x)$, jusqu’à son arrêt, produit le résultat d’une évaluation de f dotée de l’argument x . On doit aussi imposer certaines restrictions au nombre d’étapes permis et affaiblir la définition d’une fonction calculée par une machine de Turing, étant donné qu’on ne peut pas tester de façon finie jusqu’à savoir si une machine calcule une fonction pour toutes les entrées et tous les arguments.

Un castor affairé est [Rado, 1962] une machine de Turing qui atteint une activité maximale, par exemple, le nombre de pas effectués ou le nombre de symboles écrits avant son arrêt, parmi toutes les machines de Turing d’une certaine taille. Une fonction du castor affairé quantifie cette activité maximale pour une machine de Turing à n états ; ce type de fonction n’est pas calculable sauf pour de cas avec n petit.

La théorie nous dit que lorsqu’on laisse une machine tourner longtemps, la probabilité d’arrêt chute exponentiellement [Calude and Stay, 2008]. Il va toujours y avoir des choix arbitraires à faire, imposés par les restrictions dues

au problème de l'arrêt. Le choix de coupure après 1000 étapes, cependant, sort du lot à la fois à cause des valeurs connues du castor affairé à 4 états et 2 symboles (qui s'arrête après 107 étapes) et à la fois à cause des résultats dans `[FindSequenceFunction]`. Nous avons donc décidé de laisser les machines tourner environ 10 fois plus que le nombre d'étapes traversées par un castor affairé à 4 états et 2 symboles avec une entrée vide, étant donné qu'on alimente les machines avec autre chose qu'une simple entrée vide. Ces entrées sont les représentations des nombres 0, 1, ..., 20. Pour des vérifications et des recherches de convergence, la limite a parfois été fixée à un autre nombre d'étapes au cours de l'expérience (ce sera dit quand c'est le cas).

Évidemment, avec une coupure à 1000 étapes, on s'attend à ce que certaines machines s'arrêtent et d'autres non. Celles qui ne se sont pas arrêtées peuvent échouer à le faire soit parce qu'il leur faut plus de 1000 étapes pour s'arrêter, soit parce qu'elles ne s'arrêtent jamais. On a donc décidé, pour arriver à bout de ce qu'on a appelé les divergentes non-authentiques, de les associer à un programme prédicteur, qui verrait les valeurs obtenues à partir des machines s'arrêtant, et essaierait de prédire les valeurs des machines qui ne s'arrêtent pas. On dit qu'une machine diverge si, après la coupure, elle ne s'est pas arrêtée (et donc n'a convergé vers aucune valeur après ce nombre d'étapes). On appelle divergente non-authentique une machine si elle s'arrête en un temps fini pour l'entrée accordée, après la coupure des 1000 étapes. On l'appelle authentique divergente si elle ne s'arrête jamais. Comme on s'y attendait, à cause du problème de l'arrêt, on ne peut pas vraiment savoir si une série ou la valeur d'une série est une authentique ou divergente non-authentique. On peut cependant avoir une idée informelle sur le fait de savoir si une computation peut s'arrêter, en observant l'historique de calcul de la machine.

Bien que pour des raisons théoriques on ne puisse pas garantir que l'achèvement des séries à l'aide d'un programme prédicteur, soit quelque chose de parfait, on constate que l'erreur a été réduite en comparant les achèvements prévus avec des valeurs obtenues en faisant tourner les machines sur quelques étapes de plus (pour les machines qu'on soupçonne être des divergentes non-authentiques). Nous sommes conscients que des erreurs ont pu se glisser dans ces achèvements, sans qu'on puisse les éliminer. Toutefois, l'approximation à laquelle on arrive en tranchant dans le problème de l'arrêt et en faisant tourner toutes les machines jusqu'à l'arrêt, vaut mieux que de ne rien faire et se contenter de faire des comparaisons entre séries inachevées.

Comme expliqué juste avant, on va considérer que deux machines de Turing ont calculé la même fonction si (après achèvement) elles calculent les mêmes résultats sur les 21 premières entrées (de 0 à 20 en notation unaire,

même si elles divergent sur certains points), à l'intérieur des limites définies de durée d'activité. Ces résultats d'expérience sont constitués de séries de 21 valeurs, chaque valeur pour chacune des 21 entrées unitaires. Pour 21 entrées, cela signifie que 86 016 et 62 705 664 machines ont tourné sur 1000 étapes chacune pour (2,2) et (3,2), pour lesquelles un programme écrit en C tournant sur un supercalculateur avec 24 microprocesseurs a été utilisé, prenant environ 3 heures pour chaque machine pour un total de 70 heures de microprocesseur.

2.7 Le programme prédicteur

La fonction `FindSequenceFunction`, construite dans le système de calcul algébrique de *Mathematica*, propose une série de valeurs entières $\{a_1, a_2, \dots\}$ pour définir une fonction qui produit la série a_n . `FindSequenceFunction` trouve des résultats sous la forme d'une large palette de fonctions entières, ainsi que des solutions à des équations aux différences représentées par l'expression `DifferenceRoot` de *Mathematica*. `DifferenceRoot` utilise par défaut les premiers éléments de la liste pour trouver des fonctions candidates, pour valider ensuite ces fonctions en observant les derniers éléments. `DifferenceRoot` est générée par des fonctions telles que `Sum`, `RSolve` et `SeriesCoefficient`. `RSolve` peut résoudre des équations linéaires de récurrence de tout ordre à coefficients constants. `RSolve` peut aussi résoudre de nombreuses équations linéaires (jusqu'au second ordre) à coefficients non-constants, ainsi que de nombreuses équations non-linéaires. Comme ici n'est pas le lieu pour définir exactement ce que font `DifferenceRoot`, `SeriesCoefficient` ou `RSolve`, on peut imaginer le prédicteur basé sur `FindSequenceFunction` comme une boîte noire informatique spécialisée dans la prévision de séries d'entiers en se servant d'outils numériques et symboliques avancés associés à de l'analyse de régression. Evidemment, écrire un algorithme ultime pour faire des prévisions parfaites est impossible à cause du problème de l'arrêt, donc on peut seulement se rabattre sur des implémentations spécialisées limitées de ces types de programmes prédicteurs. Pour nos objectifs, `FindSequenceFunction` peut se voir comme une machine de Turing spécialisée pour prédire des séries d'entiers.

Le programme prédicteur s'empare des valeurs successives de résultat – en base 10 – d'une machine de Turing pour la série des 21 entrées en unitaire, jusqu'à tomber sur la première valeur divergente (ne s'arrêtant pas dans le temps imparti d'activité), et essaye de prédire le résultat divergent. On obtient une fonction génératrice-de-séries qui sert à complé-

ter la série s'il n'existe pas d'autres valeurs non-convergentes. Voici un exemple de série complétée de résultats de machine de Turing. Étant donné $(3, 6, 9, 12, -1, 18, 21, -1, 27, -1, 33, -1)$, on la récupère complétée en tant que $(3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36)$. A noter comment les valeurs divergentes notées -1 sont remplacées par des valeurs complétant la série grâce à l'algorithme prédicteur basé sur le `FindSequenceFunction` de *Mathematica*. Pour commencer, nous avons recherché comment de nombreuses séries étaient :

1. Pleinement complétées
2. Partiellement complétées
3. Pas complétées du tout

Et à partir de (1) et (2), quel pourcentage (a) a été correctement complété, et (b) incorrectement complété. Le seul moyen de vérifier (a) est de faire tourner la machine de Turing sur plus d'étapes. Une solution de remplacement est d'examiner la règle et d'essayer de trouver son comportement, mais ça ne marche pas toujours, et on sait qu'il est impossible qu'on trouve toujours à cause du problème de l'arrêt.

On sait donc qu'à cause du problème de l'arrêt, on ne peut pas déterminer (a) avec une absolue certitude. Cependant on veut quantifier la déviation par rapport à la certitude, et on veut vérifier si le processus parvient à compléter partiellement certaines valeurs divergentes de certaines séries. Nous avons fait tourner les machines de Turing jusqu'à 20×10^3 , et encore plus d'étapes dans certains cas, pour voir si on arrivait au bout de la computation.

On peut soulever une objection ici, en l'occurrence que la conception de la fonction pourrait favoriser certaines séries par rapport à d'autres. C'est certainement le cas, et il se pourrait que ce groupe particulier d'algorithmes soit tel qu'ils seraient incapables de compléter partiellement une série. L'objection ne serait donc pas entièrement invalide, et semblerait s'appliquer à tout montage d'une procédure de prévision. Voyons quand même ce qu'on entend par série partiellement complétée. La fonction prédicteur est définie de telle façon que, soit elle trouve la fonction génératrice-de-série, soit elle ne la trouve pas. Si elle la trouve, rien ne l'empêche de calculer n'importe quelle valeur, sauf peut-être les contraintes du hardware (*Mathematica* ne dépend que de la précision de la machine). Si la fonction ne trouve pas la fonction génératrice-de-série, alors elle ne calculera aucune valeur, ne laissant aucun espoir pour un achèvement partiel. Pour travailler sur ce problème, on s'est intéressé aussi aux séries incorrectement complétées. De ce côté, on a lancé les machines sur plus d'étapes en les comparant, puis on a introduit

le prédicteur encore une fois avec plus de valeurs. Il se peut aussi que les machines en (3,2) soient toutes computationnellement trop simples, même avec des entrées non-vides, auquel cas il faudrait entreprendre d'autres expériences plus poussées. Néanmoins, qu'elles soient trop simples ou non, on a eu huit cas en (3,2) où le programme prédicteur n'a pas pu compléter les séries. C'était les cas où les machines de Turing utilisaient les plus grandes quantités de ressources, avec les plus grandes durées d'activité et usages d'espace, comme montré dans le Tableau 3. En tout cas, le programme prédicteur nous a permis d'identifier ces cas, en échouant à les compléter.

2.8 Résultats

2.8.1 Ensembles décidables

Remarquablement, on a trouvé que seules quelques valeurs étaient nécessaires pour déterminer une série et donc décider de l'ensemble des séries générées ; on n'a pas besoin de calculer toutes les 21 valeurs dans les deux espaces (2,2) et (3,2), bien qu'on l'ait fait quand même car il n'y avait pas moyen de le savoir à l'avance.

TABLE 2.1 – Temps d'activité maximum et usages maximum de l'espace, produits par (3,2).

Machine Nb.	durée d'activité	espace utilisé
582281	8400889	4116
582263	1687273	2068
599063	894481409	27 304
1031019	2621435	56
1233829	2103293	2068
1241010	774333	1524
1233815	1687273	2068
1716199	260615	886

Les Tableaux 2.2 et 2.3 montrent que les algorithmes (constitués par les séries résultantes associées aux durées d'activité et aux usages d'espace de ruban), dans (2,2) et (3,2), sont complètement déterminés par les premières 4 et 11 valeurs des 21, ce qui signifie qu'il suffit de calculer 4 et 11 entrées pour savoir quelle fonction en (2,2) et (3,2) est en train d'être calculée. De

même, les fonctions (qui sont les séries résultantes seulement) sont définies par les 3 premières et les 8 premières valeurs de la séquence seulement.

TABLE 2.2 – Décidabilité des ensembles dans (2,2).

type de série	totalité des cas	décidables aux premières n entrées
fonctions	74	3
durées d'activité	49	3
usages d'espace	24	3
total	236	4

TABLE 2.3 – Décidabilité des ensembles dans (3,2).

type de série	totalité des cas	décidables aux premières n entrées
fonctions	3886	8
durées d'activité	3676	10
usages d'espace	763	11
total	8222	11

2.9 Classification des séries

On se sert du programme prédicteur pour essayer de prévoir les valeurs non-convergentes de la série de résultats d'une machine de Turing calculant une fonction pour les 21 valeurs, ainsi que les séries de temps d'activité et d'usage d'espace (les cellules contiguës parcourues par la tête de la machine). Avant d'entrer dans le détail des résultats, nous voudrions mentionner qu'on a découvert que les séries pourraient se répartir en six classes (pas nécessairement mutuellement exclusives). Voici quelques exemples typiques appartenant aux différentes classes. Toute-convergentes : (0, 3, 0, 15, 0, 63, 0, 255, 0, 1023, 0, 4095, 0, 16383, 0, 65535, 0, 262143, 0, 1048575, 0), (1, 6, 12, 25, 51, 103, 207, 415, 831, 1663, 3327, 6655, 13311, 26623, 53247, 106495, 212991, 425983, 851967, 1703935, 3407871). Divergentes à partir d'un certain point (sous-ensemble des authentiques divergentes) : (6,

63, 126, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1), (1, 3, 7, 15, 31, 63, 127, 25, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1). Convergentes alternant avec divergentes : (-1, 2, -1, 2, -1, 2, -1, 2, -1, 2, -1, 2, -1, 2, -1, 2, -1, 2, -1, 2, -1), -1, 0, 0, 4, -1, 20, -1, 84, -1, 340, -1, 1364, -1, 5460, -1, 21844, -1, 87380, -1, 349524, -1). Divergentes non-authentiques à partir d'un certain point, alternant parfois avec d'authentiques divergentes : (-1, 4, 16, 64, 256, 1024, 4096, 16384, 65536, 262144, 1048576, 4194304, 16777216, 67108864, 268435456, 1073741824, 4294967296, -1, -1, -1, -1), (31, 127, 511, 2047, 8191, 32767, 131071, 524287, 2097151, 8388607, 33554431, 134217727, 536870911, 2147483647, 8589934591, 34359738367, 137438953471, 549755813387, 2199023255551, 8796093022207, -1). Authentiques divergentes : (-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1), (-1, -1, -1, -1, -1, 0, 64, 192, 448, 960, 1984, 4032, 8128, 16320, 32704, 65472, 131008, 262080, 524224, 1048512, 2097088). Quelques cas difficiles à spécifier, mais appartenant définitivement à certaines des classes identifiées, vraisemblablement à la classe des non-authentiques divergentes : (0, 1, -1, 3, -1, -1, 7, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1).

TABLE 2.4 – Résumé des cas détectés (pas nécessairement mutuellement exclusifs) des fonctions calculées par toutes les machines de Turing (3,2).

Type de computation	Nb de cas	fraction
Toute-convergentes	2500	0.62
Alternance convergentes/ divergentes	383	0.095
Authentiques divergentes	1276	0.32
Alternance non-authentiques divergentes/vraies divergentes	236	0.059

Parmi les non-authentiques divergentes, 0.81% d'entre elles convergent après 20×10^3 étapes, ce qui signifie qu'on a probablement identifié toutes les divergentes non-authentiques. Finalement, parmi les machines de Turing complétées, seules huit ont résisté au second round de prédiction (ne furent pas complétées), et ont eu besoin de tourner jusqu'à 109 étapes pour s'arrêter, avec la plus grande durée avant arrêt étant exactement de 894481409 étapes < 109 .

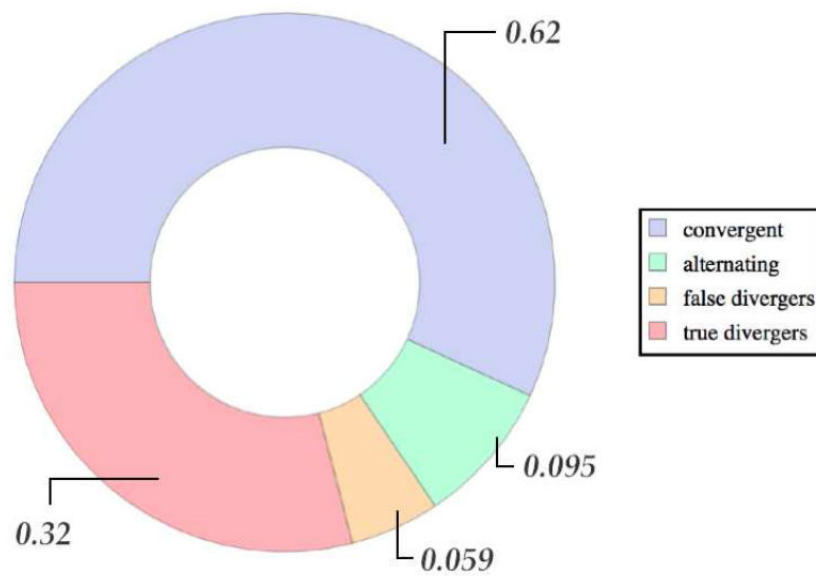


FIGURE 2.2 – Détail des cas après inspection manuelle des séries de résultats des machines de Turing, avec une limite de temps d'activité à 1000 étapes avant le processus d'achèvement (alternating signifie des valeurs divergentes et convergentes combinées).

2.10 Processus d'achèvement

2.10.1 Sélection des séries à achever (ou à compléter)

On a regardé les temps maximum d'activité de chaque machine de Turing dans (3,2) et on a sélectionné les fonctions qui ont eu des valeurs divergentes, avec un temps d'activité supérieur ou égal à 480 étapes. Parmi les 3368 machines de Turing calculées en (3,2) après 1000 étapes, on a eu 248 séries divergentes (c-à-d des machines de Turing qui ne se sont pas arrêtées pour cette durée pour au moins une des 21 entrées définissant la fonction), toutes dotées d'au moins une valeur convergente prenant au moins 480 étapes. On a choisi d'explorer cette durée d'activité car on a trouvé que les computations proches du maximum de durée (1000 étapes) tendaient à être triviales (par ex. des computations qui parcourent directement le ruban en imprimant le même symbole), et donc moins intéressantes (et moins faciles) à compléter (une chose qu'on a vérifié en échantillonnant un sous-ensemble de ces machines). De plus, après un calcul plus poussé, on a trouvé que ces computations étaient des divergentes vraies, car on les a déroulées sur 4000 étapes sans voir produites de nouvelles valeurs.

Quand on parle de prédiction, on fait référence au processus complétant les séries de résultats de fonction, de temps d'activité et d'usages de l'espace, pour une machine de Turing sur les 21 valeurs d'entrée (comme décrit en 4). Ce processus consistant à prédire toutes les séries (résultats, temps d'activités, usages de l'espace) pour une machine de Turing, est évidemment plus important que prédire une seule de ces séries (par ex. les résultats). Prédire les trois séries d'une machine de Turing pour les 21 valeurs équivaut à prédire le cheminement exact pour calculer une solution, autrement dit c'est l'algorithme exact.

Pour réduire les coûts étant donné le nombre de prédictions à faire, on a éliminé certains cas symétriques. Pour chaque machine de Turing en $(n,2)$, il en existe une autre supportant exactement la même computation (parce que les règles générées sont symétriques, et qu'elles partent des mêmes configurations initiales). Nous les appelons machines de Turing jumelles, et réduisons les données de moitié grâce à cette symétrie. Donc pour chaque paire de règles jumelles, une seule est sélectionnée. On considère ainsi 1684 séries. On a lancé le programme prédicteur basé sur `FindSequenceFunction` dans *Mathematica* pour essayer de compléter les divergentes. C'est-à-dire pour compléter les résultats, temps d'activité et usages d'espace, pour chacune des 1684 machines de Turing sélectionnées.

2.11 Comparaison entre prédictions et résultats réels

Pour qu'une prédiction soit considérée comme réussie, il faut que les séries sur les résultats, les temps d'activité et les usages d'espace coïncident pour chaque valeur avec les véritables sorties de la computation faite étape-par-étape (grâce à la vérification). On peut se retrouver dans une de ces trois situations :

La computation pas-à-pas et les séries obtenues avec le prédicteur qui complète, produisent les mêmes données, ce qui nous fait dire que la prédiction était exacte.

La computation pas-à-pas produit une valeur -1 non-convergente, signifiant par là qu'après le temps-limite, la computation pas-à-pas ne va produire aucune sortie, qui n'est pas aussi produite par le prédicteur (ce qui veut dire : ou bien la valeur nécessite une limite de temps plus importante pour être produite, ou bien le prédicteur a planté, en prévoyant une valeur convergente là où il y avait une valeur divergente).

La computation pas-à-pas produit une valeur que l'algorithme prédicteur ne prévoit pas.

Le nombre de machines de Turing incorrectement prédites fut seulement de 45 (en considérant les règles jumelles, on arrive à 90), sachant qu'on n'a pas comptabilisé comme incorrectes les séries que l'algorithme prédicteur ne pouvait pas compléter, mais qui se sont révélées convergentes. D'où, sur un total de 3368 séries complétées grâce au prédicteur, seules 90 ont été incorrectes. On peut dire alors que pour le prédicteur, la plupart des séries ont été facilement prévisibles.

En plus de ces 45 cas d'achèvement incorrect, on a trouvé 108 cas où la computation pas-à-pas produit de nouvelles valeurs que le prédicteur n'a pas prévu. Ce qui fait 153 cas en tout où des différences furent observées dans au moins une des séries qui définit un algorithme (résultats, durées d'activité, usages de l'espace), lors de la comparaison entre le programme prédicteur et la computation étape-par-étape.

2.12 Une seconde tentative de prédiction

Cette fois-ci, la computation pas-à-pas est allée jusqu'à 20×105 étapes. On a amélioré l'algorithme prédicteur pour qu'il ne se contente pas de regarder seulement les valeurs finales, mais plutôt tout l'historique de la séquence depuis son début. Maintenant, on considère que la prédiction est réussie uniquement si, pour chacune des 21 valeurs, on a : Les deux séries convergent, celle avec l'algorithme prédicteur et celle avec la computation pas-à-pas, avec les mêmes séries de résultats, de temps d'activité et d'occupations d'espace.

Les deux séries ne convergent pas, ni celles obtenues avec le prédicteur, ni celles obtenues par le pas-à-pas.

La computation pas-à-pas diverge, mais les séries obtenues avec le prédicteur ont été toutes complétées avec un déroulement $> 20 \times 105$. Il y a eu seulement huit cas d'échecs et de séries incomplètes (sans compter les règles jumelles). Ce qui fait 0.47% des machines de Turing ne pouvant pas être complétées, ce qui revient à dire qu'aucun raccourci n'a été trouvé pour elles. Ces machines de Turing non-complétées sont caractérisées par les plus grands usages d'espace et de temps, et elles se comportent comme des machines Castor Affairé [Lin and Rado, 1965] dans l'espace où on les a observées. Le tableau 4 résume nos résultats sur les processus de complétion de séries calculées dans (3,2). Ces huit séries furent finalement complétées en laissant tourner ces machines de Turing jusqu'à 109 étapes.

Seuls huit cas échouèrent à être complétés après le dernier test, et aucun d'eux ne concernait des séries incorrectement complétées. Par exemple, le prédicteur échoue à trouver la fonction générant la série de résultats : 21, 43, 1367, 2735, 1398111, 2796223, 366503875967, 733007751935, 6296488643826193618431, 12592977287652387236863, 464598858302721315448660797346840864708607 ... et n'arrive donc pas à compléter la série. La raison évidente de cet échec est le taux de croissance de la série. Les huit cas étaient tous des computations avec des valeurs de croissance super-rapides.

2.13 Expérimentations plus larges avec (4,2)

Un échantillon de 56×106 machines de Turing (4,2) fut choisi au hasard, et en ne prenant que celles dotées d'un segment initial convergent jusqu'à 1000 étapes, égal à l'une des 284 fonctions calculées en (2,2) et (3,2) (en s'appuyant sur une comparaison expérimentale concernant la complexité

temporelle, parmi différents espaces de machines de Turing ; comparaison développée en [Joosten et al., 2011]). Au final, le nombre de machines est tombé à 4 203 131. Parmi elles, 30 955 qui avaient des valeurs divergentes dont on soupçonnait qu'en fait elles étaient convergentes, furent sélectionnées pour être complétées avec le prédicteur. Une fois complétées avec le programme prédicteur, on les vérifia avec la computation pas-à-pas jusqu'à 20×10^5 étapes.

Parmi les 30 995, seules 69 machines de Turing ne furent pas complétées par le prédicteur, dont 40 s'avérèrent incorrectement complétées. Pour certaines d'entre elles, l'échec aurait pu facilement être évité avec une inspection visuelle, mais elles avaient une valeur initiale qui a trompé le prédicteur. Elles furent donc cataloguées comme des échecs stupides du programme prédicteur, dus à ses limitations particulières, plutôt que le résultat d'une série véritablement compliquée. Néanmoins, parmi les échecs, le prédicteur a incorrectement complété quelques séries d'une façon qui semblait raisonnable, mais qui s'est révélée ne pas être la vraie computation de la machine de Turing, et ce furent les cas auxquels on s'est le plus intéressé. Voici la série des temps d'activité qui fut complétée en laissant tourner la machine de Turing pendant 20×10^5 étapes : 5, 7, 19, 27, 59, 87, 179, 267, 539, 807, 1619, 2427, 4859, 7287, 14 579, 21 867, 45 925, 65 607, ... , avec 45 925 plutôt que 43 739, qui représente l'un des quelques cas où le programme prédicteur a produit une prévision incorrecte.

Pour résumer, 69 échecs sur 30 995 machines de Turing représentent un taux de réussite de 0.78 pour le prédicteur au premier passage, puis seule une toute petite fraction subsiste après un deuxième passage, en menant finalement à une prédiction totale et en confirmant donc nos observations faites dans des espaces plus petits comme (3,2). C'est en accord avec ce que nous pensons être notre découverte, qu'en l'occurrence il n'y a pas eu beaucoup de véritables fausses prédictions, et que ou bien les séries étaient trop difficiles à prédire (pas de valeurs prédites), ou bien c'était trop facile (et donc des séries incorrectement complétées).

2.14 Discussion sur l'encodage des résultats

Rappelons que nous avons adopté une représentation unaire des entrées, et que le résultat du calcul est interprété en binaire avant d'être converti en décimal. Parmi les inconvénients dus à la convention d'obtention des résultats, il se trouve que nombreuses sont les fonctions développant une croissance exponentielle (ou même plus). Par exemple le ruban-identité, c-à-d

une machine de Turing qui sort la même configuration de ruban que le ruban d'entrée, va définir la fonction $2^{n+1} - 1$. En particulier la machine de Turing qui s'arrête immédiatement en s'enfuyant hors du ruban lorsqu'elle quitte la première cellule noire (avec un 1), calcule aussi la fonction $2^{n+1} - 1$. C'est légèrement gênant, mais comme on doit le reconnaître, notre montage courant va susciter quelques occasions où on va être poussé à interpréter le résultat comme un nombre.

Pour représenter un résultat, il ne suffit pas d'utiliser uniquement des encodages/décodages qui s'arrêtent (de toute façon, tout encodage ou décodage doit toujours s'arrêter), car cette restriction ne garantit pas d'une part que le décodage/encodage puisse réaliser toutes les computations d'un système qu'on essaye de dépasser, et d'autre part n'offre aucune garantie que la représentation ne soit pas en train d'effacer ou d'ajouter de la complexité en fonction du choix de telle ou telle représentation. Il se peut qu'on se dise alors que la représentation doit être facilement calculable, ce qui peut arriver lors d'un changement de base. En tout cas, on a eu du mal à trouver un moyen simple et unique pour représenter les résultats d'une machine de Turing, car même pour le plus simple format compatible avec la série du prédicteur, l'encodage a provoqué le masquage d'une partie de la structure des computations de certaines machines, d'où un blocage du prédicteur en l'empêchant de trouver la computation – même pour ces cas simples (qui étaient simples sous leur forme originelle unitaire ou binaire).

On a trouvé intéressant et digne d'être mentionné, que le processus d'encodage, par lequel le résultat est interprété en binaire pour être converti en un nombre décimal, s'arrange concrètement dans plusieurs cas pour injecter une complexité apparente dans l'évolution de la computation originelle, en anesthésiant la fonction prédicteur qui rate alors le générateur de séquence (la formule mathématique qui génère la séquence/série) et donc n'arrive pas à dépasser la computation, à trouver de raccourci.

Pour illustrer cela, la computation de la figure 3 peut aisément se dépasser, juste en l'observant. Chaque nouvelle entrée produit une alternance de 1 et de 0, et pourtant cette série d'entrées convertie en décimal paraît plus compliquée, à cause du processus d'encodage : $s = 1, 2, 5, 10, 21, 42, 85, 170, 341, 682, 1365, 2730, 5460, 10922, 21845, 43690, 87381, 174762, 349525, 699050, 1398101$. Alors qu'en binaire, $\text{mod}_2(0 + n)$ (où mod_2 est l'opération modulo base 2) produit la série, la fonction génératrice approximative trouvée par le programme prédicteur pour cette série en nombres décimaux est $1/6(-3 + (-1)^{1+n} + 2^{2+n})$ bien qu'une représentation plus simple est possible, sous la forme d'une fonction récursive morceau par morceau :

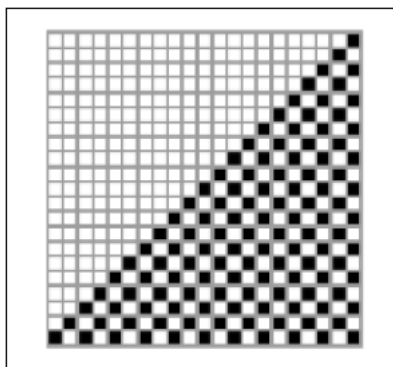


FIGURE 2.3 – Une computation d’une machine de Turing à 3 états et 2 symboles, qui paraît facile à prédire en la regardant sous sa représentation binaire originelle. Chaque rangée n est le résultat n des entrées $n = 0, \dots, 20$.

$$f(n_i) = \{(1 \text{ si } n_i = 1; 2_{i-1}^n + 1 : \text{ si } n_i \text{ pair}; 2_{i-1}^n \text{ si } n_i \text{ impair})\} \quad (2.1)$$

A noter que la fonction récursive f elle-même requiert le calcul préalable des valeurs n_{i-1} pour calculer n_i . Par définition, les fonctions récursives sont irréductibles, mais elles peuvent admettre des raccourcis – comme la formule $1/6(-3 - (-1)^n + 2^{2+n})$ trouvée quand le prédicteur a réussi à dépasser la fonction f – parce qu’elles permettent le calcul de l’élément i de la série s , sans qu’on ait besoin de calculer autre chose. La fonction récursive, dans ce cas-là, n’est pas un raccourci pour s , car elle récupère la valeur i de s sans avoir à faire tourner la machine de Turing produisant n_i . Mais à cause de la simplicité de la série, la computation de n_i demande environ n étapes, et la fonction récursive f demande n calculs. Au contraire, à la fois $\text{mod}_2(0 + n)$ et $1/6(-3 - (-1)^n + 2^{2+n})$ sont de réels raccourcis de s , même si ce dernier peut cacher la simplicité que possède la série en binaire, alors que dans le cas de la fonction récursive la simplicité est à peu près préservée malgré sa transformation en décimal.

La série de décimales est une sorte de compilateur entre le langage résultant de la machine de Turing (base 2) et le langage du programme prédicteur (base 10). Etant donnée la façon dont le programme prédicteur fonctionne, basée sur la fonction `FindSequenceFunction` de *Mathematica*, il ne peut prendre comme entrée qu’une série de nombres entiers constituant un argument.

On peut injecter ou masquer de la complexité apparente quand on trans-

forme une représentation numérale en une autre. Pour que le lecteur de sortie voie des motifs, il doit être capable de lire la sortie dans le langage du système originel (le binaire, dans ce cas) sans le traduire. Il n'est même pas sûr que l'exploration des motifs dans d'autres bases nous dise quelque chose sur les motifs de la séquence originelle.

Nous pensons qu'à cause de ce genre de trouvaille intéressante, ces questions mériteraient une étude plus approfondie. Nous avons découvert par exemple d'autres phénomènes artificiels, tels que des "phases de transition" (changement évident de vitesse) dans la distribution des durées avant arrêt [Joosten et al., 2011], plus dues à ces conventions qu'à de véritables propriétés du système étudié.

Troisième partie

Vers une théorie algorithmique du monde

Chapitre 3

Des univers computables

Depuis les époques de Newton et Leibniz, les scientifiques ont construit des visions élaborées du monde. Au siècle dernier, la mécanique quantique révolutionna notre compréhension de la réalité physique à l'échelle atomique, tandis que la relativité générale faisait de même pour notre compréhension de la réalité à une plus grande échelle.

Certaines visions contemporaines du monde ont une approche des objets et des lois physiques, en termes d'information et de computation, auxquels elles assignent une responsabilité ultime pour la complexité de notre monde, incluant la responsabilité pour des mécanismes complexes et des phénomènes tels que la vie. Dans cette vision, l'univers et les choses en lui sont vus comme se calculant eux-mêmes. Nos ordinateurs ne font que reprogrammer une partie de l'univers pour faire en sorte qu'il calcule ce qu'on veut qu'il calcule.

Certains auteurs ont élargi la définition de la computation aux objets physiques et aux processus physiques, à différents niveaux de la réalité physique, allant du digital au quantique. La plupart des penseurs de pointe impliqués ont été des contributeurs de mon recueil [Zenil, 2012a], dont certains qui s'opposent à l'approche digitale ou calculable.

La vision computationnelle/informationnelle (parfois identifiée comme *computationnalisme*) se base sur les réflexions de pionniers comme John A. Wheeler et Richard Feynman. Une citation tirée des Messenger Lectures de Feynman, délivrée en 1964 à la Cornell University, distille en substance que la nature doit probablement opérer à un niveau très simple, malgré son apparente complexité pour nous, et marque un changement dans notre compréhension de la physique en termes d'information digitale.

Cela m'a toujours ennuyé que, selon les lois telles que nous les

comprenons aujourd’hui, une machine calculante ait besoin d’un nombre infini d’opérations logiques pour trouver ce qui se passe quelque part, quelle que soit la petitesse de la région de l’espace et quelle que soit la petitesse de l’intervalle de temps. Ainsi, j’ai souvent caressé l’hypothèse que la physique va finir par ne plus avoir besoin de déclarations mathématiques, qu’à la fin la machinerie va être mise à jour, et que les lois vont s’avérer toutes simples, comme l’échiquier avec ses complexités toutes apparentes.

Cette vision fut probablement influencée par son directeur de thèse John A. Wheeler, l’auteur de l’expression “it from bit”, suggérant que l’information constitue le niveau le plus fondamental de la réalité physique. Un autre pionnier fut Konrad Zuse, dont j’ai le plaisir de pouvoir présenter en Appendice de cette thèse pour la première fois une nouvelle édition de son *Espace Calculateur* (Rechnender Raum). Avec Adrian German (Université de Indiana, Bloomington) nous avons fait de gros efforts pour traduire en LaTeX une version ancienne scannée de la traduction originale commandée par Ed Fredkin et publiée par le MIT. Fredkin est lui-même un autre pionnier, ayant fondé le champ connu aujourd’hui sous le nom de physique digitale, et il est un des contributeurs de mon recueil [Zenil, 2012a]. Stephen Wolfram, un étudiant de Feynman, s’est aussi construit selon cette vision, étant le fer de lance d’un changement facilité aujourd’hui par la disponibilité de ressources de calcul de plus en plus grandes et de moins en moins coûteuses. Encore plus récemment, certaines théories modernes de la physique, en développement, ont essayé d’unifier mécanique quantique et relativité générale, sur la base de l’information (’t Hooft, Susskind, Smolin), un développement représenté dans le travail de certains contributeurs du débat (voir Appendice).

Ces visions contemporaines d’un univers computationnel sont aussi profondément liées, via le concept de l’information, à un champ contemporain de recherche mathématique appelé la théorie algorithmique de l’information (AIT). Les chercheurs de l’AIT pensent que la vraie nature de la nature ne peut être révélée que par l’étude de la notion d’aléatoire (Greg Chaitin, Leonid Levin, Cris Calude).

3.1 Qu’est-ce que la computation ? Comment calcule la nature ?

Zuse a suggéré assez tôt que le monde pouvait être le résultat d’une computation digitale déterministe. Fredkin a développé plus tard l’idée que

l'univers pourrait être décrit par un automate cellulaire (voir sa contribution dans [Zenil, 2012a]). Dans son approche minimaliste, Wolfram [Zenil, 2012a] défend l'idée que le monde pourrait s'avérer être le résultat de très simples règles, peut-être même d'une seule, à partir de laquelle a émergé l'apparente complexité visible autour de nous.

Si l'univers est un univers computable, il suffirait de lancer un calculateur universel de Turing sur tous les programmes possibles pour générer non seulement notre propre univers, mais tous les univers possibles, comme Wolfram et Schmidhuber [Zenil, 2012a] l'ont suggéré (tous deux sont contributeurs de mon recueil [Zenil, 2012a]). La question serait alors de savoir comment distinguer notre univers de tous les autres univers possibles. Comme Wolfram l'a souligné, il y aura certains univers qui seront clairement différents du nôtre, et de nombreux autres qui y ressembleront beaucoup, auquel cas on peut se demander si le nôtre va se trouver être spécial ou peu commun d'une façon ou d'une autre, et si par exemple, il se situerait parmi les premiers en terme de taille nécessaire pour le décrire, c.-à-d. parmi ceux qui possèdent la description la plus courte. S'il est suffisamment simple, l'AIT laisserait alors supposer aussi qu'il n'est pas rare, le résultat de nombreux programmes générant dans ce cas le même univers. C'est une déduction basée sur les probabilités algorithmiques, qui décrivent la distribution des structures, et qui établissent une connexion forte avec la complexité algorithmique.

Selon l'approche de Schmidhuber, il semblerait qu'un ordinateur générant tous les univers possibles doive nécessairement être plusieurs fois plus grand que l'univers lui-même. Mais si les programmes sont assez courts, au lieu de lancer tous les programmes étapes par étapes, il serait possible de commencer avec certains univers plausibles et les lancer pendant plus longtemps, comme Wolfram l'a suggéré, en vérifiant chaque programme et en laissant se déployer ceux qui semblent complexes.

Il y a aussi ceux qui pensent que la nature ne réalise pas seulement une computation digitale (comme le voient Zuse, Fredkin ou Wolfram, tous contributeurs de mon ouvrage [Zenil, 2012a]), mais qu'elle est elle-même le résultat d'une computation quantique (Lloyd, Deutsch et Cabello, aussi auteurs de chapitres dans cet ouvrage). Selon eux, le monde se baserait en dernière instance sur la physique, particulièrement sur la mécanique quantique, et serait le reflet des propriétés de particules élémentaires et de forces fondamentales. Lloyd [Zenil, 2012c] par exemple, se demande de combien de bits se compose l'univers, offrant un intéressant calcul selon lequel, étant donné les propriétés des particules quantiques, l'univers ne peut être représenté par une description plus courte que lui-même—simplement parce que toutes les particules élémentaires auraient besoin d'être simulées par une autre par-

ticule élémentaire. Il faudrait que l'ordinateur simulant l'univers soit de la taille de l'univers, et cela nécessiterait l'énergie de l'univers réel, le rendant indiscernable de cet ordinateur (quantique), le calcul et l'ordinateur étant en parfaite correspondance.

Si le but n'est pas de décrire avec la plus grande précision notre univers dans l'état où il est, on peut quand même se demander si un univers de complexité semblable nécessiterait une description très compliquée. Le débat semble donc tourner autour d'une description possible de l'univers, si elle peut s'écrire en bits ou en qubits, si elle peut être plus courte que l'univers lui-même ou non. Cependant, si l'information digitale se situe en amont du quantique, il pourrait s'avérer que la plus courte description de l'univers soit beaucoup plus courte que l'univers lui-même, contrairement aux visions de Lloyd, Deutsch, Cabello, Calude et al., par exemple, et qu'elle puisse être compressée dans un simple programme court de calcul, comme Zuse, Wolfram, Fredkin, Bolognesi et moi-même le pensons.

En fait, on peut concevoir le but de la physique digitale en tant que modèle minimal décrivant l'univers, équivalent au but de la physique dans sa quête pour une théorie unifiée, cherchant une formule descriptive pour l'organisation de toutes les forces et particules encore à découvrir, s'il en reste. Qu'un tel programme existe est une question ouverte, tout comme c'est une question ouverte de savoir s'il existe une théorie du tout (ToE). Mais ce n'est plus une question ouverte de savoir si une telle unification peut se faire pour de très grandes portions de la physique. C'est un fait accompli pour des champs tels que gravitation et mouvement, électricité et magnétisme, électromagnétisme et le plupart des forces électronucléaires, des domaines de la physique aujourd'hui modélisant avec une précision étonnante de larges portions de la nature, en utilisant des lois simples pouvant être programmées sur un ordinateur, et pouvant en principe (et souvent de facto) fournir de parfaites prescriptions et prévisions sur le monde. En fait, on dispose de lois physiques et de programmes informatiques pour à peu près tout ; ce qui nous manque, c'est une théorie unique qui embrasserait toutes les autres théories.

3.2 L'approche algorithmique

Si le monde, en fait, n'est pas un "calculateur digital", il pourrait cependant se comporter comme tel. Donc, qu'il soit ou non un ordinateur, on pourrait essayer de tester si les résultats des processus dans le monde ressemblent aux résultats auxquels on pourrait s'attendre en faisant tourner un programme. Il existe de nombreuses lois qu'un ordinateur peut suivre.

Ou si vous préférez, les ordinateurs suivent un sous-ensemble spécifique de lois physiques associées au traitement de l'information, notamment la distribution de patterns décrite par les probabilités algorithmiques. Selon moi, l'information ne peut exister dans notre monde que si elle est supportée par un processus ; chaque bit doit avoir son porteur physique lui correspondant. Même si ce porteur n'est pas matériel, il prend la forme d'une interaction entre composants matériels—un atome interagissant avec un autre atome, ou une particule avec une autre. Au niveau le plus bas, cependant, la plus élémentaire des particules, tout comme les bits isolés, n'est porteuse d'aucune information (l'entropie de Shannon d'un bit unique est 0 car on ne peut implémenter aucun canal de communication avec un seul bit, 1 et 0 étant dotés du même contenu informatif possible, si pris isolément). Les particules isolées peuvent n'avoir aucune histoire causale, étant sans mémoire quand elles sont isolées de toute interaction extérieure. Quand des particules interagissent avec d'autres particules, elles s'attachent à un réseau causal et semblent obligées de définir une valeur comme résultat de cette interaction (par exemple une mesure). Ce qui nous surprend au sujet du monde quantique, c'est justement son manque de causalité apparente, chose qu'on voit partout ailleurs et dont nous avons l'habitude. Mais c'est l'interaction et son histoire causale qui supportent toute la mémoire du système, avec le nouveau bit apparaissant comme s'il avait été défini au hasard, car nos théories de la mécanique quantique fournissent seulement des amplitudes de probabilités. Lier un bit à un réseau causal peut sembler équivalent à produire une corrélation de mesures entre des parties de l'espace apparemment disjointes, alors qu'en fait elles pourraient bien avoir été connectées depuis toujours, si le monde est pris comme déterministe (une vision implicite dans les modèles dits à variables cachées). La distribution universelle de Levin [[Kirchherr et al., 1997](#)], basée sur les probabilités algorithmiques, décrit les fréquences de sorties attendues en fonction des complexités de ces sorties. Un processus produisant une chaîne s avec un programme p , quand il est exécuté sur une machine de Turing T , possède la probabilité $Pr_{T(s)} = 2^{-|p|}$, où $|p|$ est la longueur du programme p . Le théorème du codage [[Calude, 2002](#), [Cover and Thomas, 2006](#)] fait la liaison entre la fréquence $Pr(s)$ avec laquelle une chaîne s est produite, et sa complexité algorithmique $C(s)$. Ce qu'on appelle la semi-mesure m possède aussi la remarquable propriété de dominer Pr_T pour toute machine de Turing universelle T . En substance, $m(s)$ établit que s'il y a beaucoup de descriptions longues d'une certaine chaîne, alors il existe aussi une description courte dotée d'une faible complexité algorithmique $C(s)$, de sorte que $m(s) \sim 2^{-C(s)}$. Comme ni $C(s)$ ni $m(s)$ ne sont calculables, il ne peut exister de programmes prenant une chaîne s comme entrée pour produire en sortie $m(s)$. Cependant, nous avons prouvé [[Delahaye and Zenil, 2012](#)] que des approximations numé-

riques sont possibles et que des évaluations numériques raisonnables donnent des résultats raisonnables. Tout comme les chaînes peuvent être produites par des programmes, on peut se demander, selon les probabilités d'apparition d'un certain résultat issu d'un certain phénomène naturel, si ce phénomène, tout comme une machine calculante, n'est pas un processus plutôt qu'un événement aléatoire. Si on ne dispose d'aucune autre information sur le phénomène, on peut voir si $m(s)$ nous dit quelque chose au sujet d'une distribution des résultats possibles dans le monde réel [Delahaye and Zenil, 2011]. Dans un monde de processus calculables, $m(s)$ indiquerait la probabilité qu'un phénomène naturel produise un résultat particulier, et nous dirait si une certaine structure va se produire souvent ou non. Considérons une opération inconnue générant une chaîne binaire de longueur k bits. Si la méthode est uniformément aléatoire, la probabilité de trouver une chaîne particulière s est exactement 2^{-k} , la même pour toute chaîne de longueur k , ce qui vaut pour les chances de prélever les digits de π au hasard. Néanmoins, les données (tout comme π —largement présent dans la nature, par exemple sous la forme de processus courants associés à des courbes) sont habituellement produites non pas au hasard, mais selon un processus spécifique (dans le cas de π , il existe de nombreuses façons de le produire; il y a même des phénomènes physiques qui mènent à lui). C'est là où $m(s)$ peut être pertinent quant au calcul de la probabilité de processus physiques.

3.3 La complexité des systèmes vivants

La biologie a été témoin d'une transformation durant le siècle dernier, qui a commencée avec les découvertes de Mendel concernant le transfert de certains traits de caractères chez les pois, un phénomène relevant d'un transfert d'information entre générations. Redécouvertes sur le tard, ses lois ont posé les fondements de ce qui est aujourd'hui la génétique moderne, une science qui a établi que les organismes vivants stockent et lèguent de l'information comprenant des instructions pour leur développement complet, codée—comme Watson et Crick l'ont découvert—dans les acides ribonucléiques et nucléiques. Le code de la vie est digital; deux bits par paire de base sont nécessaires pour encoder l'ADN.

Les règles déterminant la façon dont l'ADN se réplique sont probablement algorithmiques par nature, comme celles qui gouvernent d'autres types de phénomènes physiques, et nous conduisent parfois à découvrir de fortes ressemblances dans les distributions de motifs produits. Les processus concernant l'ADN sont relativement simples. Un sous-ensemble d'opérations pure-

ment digitales peuvent correspondre à des opérations computationnelles, des opérations telles que la jointure, copie, partition, complémentation, coupe, ou remplacement. Ce qui implique que, couche après couche, le code de la vie s'est construit sur des milliards d'années au cours d'un profond processus algorithmique doté de ses propres règles caractéristiques, faisant en sorte que des processus comme le déploiement des protéines nous apparaissent hautement complexes. Si nous ignorons l'algorithme du déploiement des protéines, il n'y a pas de raison de penser que le déploiement d'une protéine ne puisse pas être supporté par une (déterministe) machinerie (que ce soit en temps polynomial ou non).

Dans le cas de structures dont l'état final est assuré (que ce soit une protéine déployée ou la division d'une cellule), la solution est soit le résultat d'un algorithme, soit un processus aléatoire. Il est peu probable que le déploiement d'une protéine soit aléatoire, car une protéine incorrectement déployée causerait sûrement une maladie. Une des propriétés les plus importantes de la vie est la robustesse, dont nous pensons qu'elle peut s'expliquer par les probabilités algorithmiques [Zenil and Marshall, 2013].

Si la computation est la force conductrice produisant les structures dans le monde, on peut s'en servir de base pour manipuler l'orientation de la complexité. On pense au problème du rassemblement des bons éléments chimiques pour obtenir la vie. Si on s'attend à ce que la vie émerge de constituants chimiques, si on attend d'eux la production, par exemple, des digits de la constante mathématique π , les chances que cela arrive sont ridiculement minces. Comment la nature a-t-elle produit des organismes d'une telle complexité ? Wolfram, par exemple, pense que la nature puise dans ce qu'il appelle l'univers computationnel des programmes possibles. Le concept d'évolution darwinienne peut nous laisser supposer que, quels que soient les processus qui ont donné naissance aux formes observées dans les systèmes biologiques, ils doivent avoir été complètement dessinés par la sélection naturelle. Mais ils existent de forts indices (par exemple [Wolfram, 2002]) que la nature échantillonne des programmes, que la nature dispose, d'une certaine façon, de ce que la computation propose. Si maintenant on se repose la même question sur la production de patterns, dans le contexte de la computation, la probabilité d'auto-assemblage chimique pour produire les digits de π par hasard est beaucoup plus grande, car il suffit de trouver un programme produisant π plutôt que les digits eux-mêmes. Les programmes produisant π sont infiniment plus courts que le nombre des digits de π .

Les patterns naturels peuvent aussi être le résultat d'un même genre de formation raccourcie, et la cause des structures visibles dans la nature. Ces patterns ne se répètent pas en se copiant eux-mêmes ; ils sont récurrents car

ils sont le résultat de programmes simples produisant le même pattern encore et encore, dont la fréquence est dictée par la probabilité algorithmique. Que le pattern doive changer, cela signifie que la règle a changé aussi, mais comme le prévoit la probabilité algorithmique, si le programme reste court, ses chances de produire un pattern de faible complexité algorithmique, peut-être même le pattern semblable à celui produit avant la mutation, sont exponentiellement grandes comparées aux chances qu'il a de produire des patterns d'apparence aléatoire. Il serait alors plus logique de penser que la sélection naturelle se situe au niveau des programmes, de façon à préserver la propriété fondamentale de résilience des structures vivantes, c.-à-d. la robustesse. Donc les patterns sont robustes car ils résultent d'une règle courte, concise.

D'un autre côté, si on était libre de sélectionner des motifs/patterns aléatoires plutôt que des programmes aléatoires, les résultats seraient plutôt désordonnés et d'apparence aléatoires. Et ce n'est pas ce qui semble se passer, c'est pourquoi on ne voit pas de soudains changements dans la nature, même si on observe un certain aléatoire apparent. Une question à se poser légitimement est le rôle de cet aléatoire apparent en biologie, si c'est lui qui conduit aux spéciations biologiques basées sur les mutations. Est-ce que c'est un aléatoire seulement en apparence ? Finalement, le monde est-il plus aléatoire que structuré (voir par exemple Calude. Un autre partisan de cette vision est McAllister [[McAllister, 2003](#)]).

Comment le monde pourrait-il être un simple programme d'ordinateur ? Si tous les processus physiques sont eux-mêmes computables, Turing a montré qu'il existe une machine unique capable de tous les faire tourner. Dit autrement, on peut construire un programme unique parmi tant d'autres. Bien sûr, prétendre que les processus ou le monde lui-même puissent être un programme informatique ne signifie pas qu'ils le sont réellement, ou qu'ils se comportent comme des machines de Turing. Nous ne devons pas nous leurrer en pensant que la proposition disant que la nature est computable signifierait que l'univers est une machine de Turing. Il existe évidemment des processus naturels qui ne sont définitivement pas comme des machines de Turing. La machine de Turing est une sursimplification du concept de computation (digitale). Le cerveau humain, par exemple, est un objet très différent d'une machine de Turing, malgré les parallèles existant entre eux. La question de savoir si (tous) les processus physiques opérant dans le cerveau humain peuvent être supportés par une machine de Turing ne doit pas être prise comme une suggestion que le cerveau est une machine de Turing. Dans mon recueil [[Zenil, 2012a](#)], Szudizk et Hutter fournissent des formalismes mathématiques précis pour l'hypothèse d'un univers computable.

Mais le monde est dominé par des processus computables, d'un type qui

peut être supporté par de la computation digitale, donc la plupart des structures du monde peuvent être mises au seul crédit de la computation, car la computation suit la distribution universelle de Levin au sujet de la fréquence d'apparition des patterns, que décrit la probabilité algorithmique. Le fait que, malgré leurs limitations, les modèles scientifiques puissent décrire une grande partie du monde, et que de tels modèles soient computables, suggère fortement que, bien que la nature puisse aller au-delà de la computation, son activité dépend largement de la computation (de Turing). Au cours de l'histoire, nous nous sommes arrangés pour que la nature fasse ce qu'on veut qu'elle fasse. Le processus a commencé avec de simples outils, pour finalement nous conduire aux ordinateurs programmables pour réaliser toutes sortes de calculs en notre faveur, et utilisés pour fabriquer d'autres appareils faisant toutes sortes de travaux, en guise d'outils. Si la nature fait plus que juste calculer, nous savons néanmoins qu'elle est capable de calculer, et même très bien. On peut avancer un argument contre cette vision, c'est l'incapacité des modèles scientifiques à faire des prévisions avec une précision arbitraire. Mais leur non-linéarité n'implique en aucune façon un indéterminisme ou une non-calculabilité. Comme Wolfram l'a montré, même le plus simple programme informatique (tel que l'automate cellulaire élémentaire (ECA) de règle 30) est très difficile, sinon impossible à comprendre assez bien pour être capable de prévoir la manière dont il se déploie—malgré sa désarmante simplicité.

Nous n'avons pas beaucoup de raisons, peut-être aucune, de croire qu'il existe quelque chose d'inhérent dans la nature qui ne soit pas Turing-calculable, même si imprévisible. Le fait qu'on soit continuellement capable de remplacer les théories par d'autres toujours plus fidèles et précises, peut se voir comme un processus computable en lui-même, au sens même de la définition de calculabilité de Turing. Pour se servir de π encore une fois, π est un nombre computable car une machine de Turing peut calculer n'importe quel chiffre du développement digital de π . Donc même si nous ne sommes pas capables d'offrir et calculer le développement complet de π en toutes bases, le fait qu'il soit calculable veut dire qu'on peut toujours en fournir une meilleure approximation. Pour tout digit indexé i dans l'expansion de π , on peut toujours calculer $i + 1$ digit de π . Le nombre π est aussi d'apparence aléatoire et imprévisible à bien des égards, car nous ne pouvons dire quel digit va apparaître sans faire un calcul. Même s'il existe des formules [Bailey et al., 1997] pour calculer arbitrairement les digits de π en bases $2n$, qui ne nécessitent pas de calculer les digits précédents, le résultat du calcul d'un digit de π ne le rend pas plus facile à reconnaître. Les digits de π , dans toutes les bases dans lesquelles il a été étudié, semblent aléatoires selon tous les tests statistiques courants, et bien qu'il n'y ait aucune preuve de sa normalité, π est considéré

comme tel, ceci signifiant qu'il possède une propriété d'un véritable nombre aléatoire, c.-à-d. qu'il contient tous les motifs possibles, bien qu'il ne soit pas un motif lui-même, étant aléatoire. On ne peut exclure la possibilité de trouver des formules du même genre ou d'un genre complètement différent, qui permettent néanmoins une rapide computation de digits individuels de l'évolution de ces systèmes apparemment aléatoires.

3.4 Comment s'intègre le Quantique

Contrairement à l'apparence d'aléatoire discutée dans la section précédente au sujet de π ou de l'ECA de règle 30, les événements dans la mécanique quantique semblent survenir sans raison apparente—par exemple le moment où une particule va se désintégrer ou la position d'un électron s'effondrant en s'affranchissant d'une intrication. De façon assez surprenante, les idées les plus sérieuses sur le caractère digital de l'univers viennent de la mécanique quantique (enfonçant leurs racines dans une longue tradition initiée par les Grecs anciens et leur concept d'atomisme). Ce fut Max Planck, en essayant de comprendre l'émission radiative de corps chauds, qui découvrit le phénomène de quantisation de la radiation. La mécanique quantique est effectivement conçue comme étant digitale, mais elle est en même temps analogique. Un électron peut être simultanément en plusieurs endroits, couvrant une région étendue de l'espace. Ainsi la mécanique quantique n'a pas réglé le problème une fois pour toutes, car les particules se comportent elles-mêmes comme des ondes, préservant ainsi une dualité entre les descriptions digitales et analogiques des blocs de construction les plus fondamentaux de l'univers (pour un débat sur ce sujet, voir [Zenil, 2011]).

Ce n'est pas une question de philosophie, ni même de mathématique, c'est une question de physique, avec réponse définitive. Si on zoome assez profondément à l'intérieur de la réalité, et si le monde est digital, on devrait pouvoir observer des pixels et des bits. Cependant, si le monde n'est pas digital, on pourra toujours zoomer sans être capable de prouver ou d'infirmer toute possibilité. Une troisième réponse est que ce n'est ni l'une ni l'autre, impliquant que c'est concrètement les deux, juste comme le fait l'interprétation standard de la mécanique quantique.

Il existe toutefois une tendance qui a favorisé une discrétisation des modèles (pour certaines discussions à ce sujet, voir [Zenil, 2012b]). On pensa d'abord que l'électricité était continue, puis l'électron fut découvert. Comme avec le photon, la gravitation suit aujourd'hui une approche du même genre. Il semblerait qu'on passe d'une théorie continue à une théorie atomique, pour

en finir avec une réponse non satisfaisante (la dualité des particules a conduit à une explication de l'électricité en termes d'électrons, qui eux-mêmes se comportent comme des particules et des ondes). Dans ce processus, on en est arrivé à réaliser que l'information jouait un grand rôle, lorsqu'on s'est aperçu que ce que font beaucoup de ces fusions théoriques quant aux objets et phénomènes, c'est de les dépouiller de certaines caractéristiques assumées au départ leur être exclusives, pour les attribuer à un objet ou un phénomène composite dont on considère qu'ils font maintenant partie.

Au début du siècle dernier, le développement de la mécanique quantique a poussé les physiciens à repenser radicalement les concepts qu'ils utilisaient pour décrire le monde. Aucun compte-rendu de ce que pourrait être l'information, ne peut se considérer comme complet s'il ne prend pas en compte les interprétations de la mécanique quantique. Les systèmes classiques s'accordent avec le critère de ce qui peut s'identifier comme un réalisme local, qui est que les résultats de mesures d'un système localisé dans l'espace-temps sont totalement déterminés par des propriétés inhérentes à ce système (sa réalité physique), et ne peuvent pas être instantanément influencés par un événement distant (localité). En d'autres mots, la localité interdit toute influence entre événements situés dans des régions dites spatialement séparées, tandis que le réalisme prétend que tous les résultats des mesures dépendent de propriétés pré-existantes des objets, qui sont indépendantes de la mesure. Les phénomènes quantiques, cependant, semblent fondamentalement non-locaux au sens relativiste, lorsqu'on est obligé d'invoquer un cadre particulier de référence pour donner un sens à la déclaration disant que la mesure d'une particule survient d'abord, puis que son résultat affecte immédiatement l'état d'une seconde particule, même si elle se situe hors des limites permises pour l'échange d'information entre les deux par la théorie de la relativité.

Comme découvert par Albert Einstein, Boris Podolsky et Nathan Rosen, la mécanique quantique prédit de fortes corrélations entre des mesures portant sur deux particules en état intriqué. Il est tentant d'interpréter ces corrélations comme le résultat de propriétés partagées, déterminées au moment de leur interaction initiale puis assimilées dans chaque particule. La formulation par Bell d'une inégalité⁵ a permis de calmer le débat, autorisant la réalisation d'une expérimentation pour tester les inégalités et montrer que la théorie des variables cachées, basée sur la supposition associée de localité et de réalisme, était en désaccord avec les prévisions statistiques de la mécanique quantique.

Depuis lors, les physiciens ont monté des expériences pour vérifier la réalité quantique. La première fut celle de John Bell lui-même, qui montra que si on supposait que les particules étaient corrélées, au sens de mesure des pro-

priétés de l'une donnant les propriétés de l'autre, alors il existait une inégalité qui décrivait le maximum de corrélation possible dans un monde classique. Plus tard, des raffinements ont continuellement convergé vers les tests d'inégalités de Bell, fermant à la fois l'échappatoire de la localité et l'échappatoire de la détection. Donc les scientifiques du fondement quantique pensent qu'il est raisonnable de considérer la violation du réalisme local comme un fait bien établi.

Pour que l'expérience de Bell valide la réalité quantique, il faut la concevoir pour couvrir toutes les portes ouvertes ou échappatoires possibles. Se pourrait-il qu'il y ait un certain biais induit par les expériences de Bell qui fasse que ce soit les particules qui sont le plus corrélées qui aient le plus de chances d'être mesurées? Cela expliquerait la violation. Alain Aspect [[Aspect et al., 1982](#)] conçut une expérience pour vérifier et exclure cette possibilité. Elle impliquait l'usage de plusieurs détecteurs mesurant un grand nombre de paires de photons pour obtenir des résultats statistiquement significatifs. Après plusieurs expérimentations, la communauté fut convaincue que les expériences d'Aspect avaient exclu la possibilité que l'expérience de Bell soit statistiquement biaisée et ne mesure que les particules corrélées. Se pourrait-il que cela et d'autres expérimentations soient toujours en train de nous tromper, nous faisant croire que la réalité quantique se comporte d'une certaine façon alors que ce n'est pas le cas?

Aspect a contribué à écarter le problème concernant un biais local dans la mesure de particules corrélées, mais qu'en est-il si quelque chose influençait l'expérience en communiquant des propriétés entre les particules? Une façon d'exclure cette possibilité est de s'assurer d'une distance suffisante pour garantir, grâce à la vitesse de la lumière, que si les deux mesures aux deux extrémités étaient assez rapides, rien ne pourrait voyager pour communiquer quoi que ce soit d'une particule à l'autre. La supposition ici est que même lorsque les deux particules sont corrélées car elles viennent d'une même source, elles ne devraient pas être corrélées au-delà de ce que l'inégalité de Bell établit, et que si une corrélation est supérieure à cela, ça signifie qu'il existe une inquiétante intrication entre les particules quantiques conduisant l'une d'elle à être au courant ou à modifier ses propriétés dès que l'autre a été mesurée.

Si deux de ces particules sont assez proches pour communiquer entre elles, la violation des inégalités de Bell est facilement explicable. Il faut donc garantir que l'information ne puisse pas être disséminée avec l'aide d'un porteur physique, un porteur matériel qui ne peut voyager plus vite que la vitesse de la lumière. Donc en plaçant deux particules suffisamment loin l'une de l'autre, on peut essayer d'exclure cette possibilité (encore une fois, si on n'est pas dans le cas où il existe une couche d'information sous-jacente, conte-

nant une réalité à variables cachées ou communiquant les valeurs mesurées). La disponibilité de sources hautement efficaces de particules intriquées (avec l'émergence de la technologie laser) a permis aux scientifiques quantistes de réaliser des *expériences d'Aspect* sur une distance suffisamment grande pour apparemment fermer cet échappatoire, empêchant les particules de communiquer entre elles sans avoir à supposer qu'elles l'ont fait plus vite que la vitesse de la lumière, ce que la relativité générale rend hors de question.

Cependant, cela ne peut marcher que si le réglage des détecteurs est continuellement modifié, étant donné que les expériences précédentes peuvent les avoir influencés, ou qu'un détecteur ait pu trouver quel était le réglage de l'autre. Pour garantir la validité du résultat des mesures de corrélation, il ne doit y avoir aucune mesure corrélée au départ. Weihs et al. [Weihs et al., 1998] ont réussi à séparer de 400 mètres deux particules, ce qui leur a donné 1,3 microsecondes pour changer aléatoirement le réglage des détecteurs, et un maximum de 5 nanosecondes entre les mesures, garantissant qu'à la vitesse de la lumière, aucune information ne pouvait être transmise entre les deux extrémités. Même si on a fermé certains échappatoires dans les tests de la mécanique quantique, d'autres peuvent s'ouvrir (par exemple l'effondrement de localité), et aucun test n'a encore englobé tous les échappatoires en même temps. La théorie réaliste-locale à variables cachées, préféré par Einstein, peut ne pas être une description viable du monde, mais il y a d'autres échappatoires à fermer, surtout celui qui suppose et conduit finalement à un argument circulaire.

Il se trouve que la tentative susmentionnée pour clore un échappatoire, suppose la capacité à modifier aléatoirement le réglage des détecteurs des deux côtés, en tenant leurs générateurs aléatoires assez éloignés l'un de l'autre pour s'assurer que le choix d'un générateur n'influence pas l'autre. C'est un échappatoire qu'aucun physicien expérimental ne pense digne de tenter de fermer, car il y a peu de choses à tester, sinon rien. Il se trouve que tous les tests sur les inégalités de Bell supposent qu'on est libre de choisir les réglages des détecteurs, et qu'une telle liberté doit être indéterministe pour que l'interprétation des résultats de l'expérience fonctionne. Mais en pratique, on est obligé de choisir entre deux options : soit on utilise un générateur pseudo-aléatoire, ce qui signifie que la source d'aléatoire n'est pas vraiment aléatoire mais déterministe et donc potentiellement corrélée aux réglages des autres générateurs, soit on est libre de choisir à partir d'un ensemble d'opérateurs non commutables, c.-à-d. de commencer avec du libre arbitre et du véritable aléatoire (indéterministe) lors de la préparation de l'expérience.

Cependant, si l'univers se déroule de façon déterministe, il n'y a rien à expliquer et toutes les étrangetés quantiques prennent tout de suite un sens.

C'est en accord avec ce que Bell lui-même reconnaissait comme une possibilité : si la mécanique quantique est totalement déterministe, alors même la décision de l'expérimentateur de mesurer certains composants des spins est entièrement déterminée, ainsi l'observateur ne pourrait pas avoir décidé de mesurer autre chose que ce qu'il mesure de fait. Autrement dit, les corrélations auraient une explication triviale, car il est trivial de penser à des états corrélés à tous les niveaux, étant donné que tout l'univers serait l'effet d'un état antérieur et partagerait donc simplement une origine commune. Par analogie, pensons aux ensembles semblables de chromosomes entre frères et sœurs. Le partage des gènes explique les corrélations dans la couleur des yeux et autres caractères, sans avoir à introduire d'autres explications tortueuses. Que tout l'univers soit déterminé depuis le tout début, c'est bien sûr un fort engagement à prendre pour rendre tout son sens aux expérimentations quantiques, mais c'est sans doute beaucoup plus simple que la plupart, sinon toutes, des interprétations courantes de la mécanique quantique.

Il doit y avoir une raison pour essayer de passer en force une interprétation de la mécanique quantique où le seul résultat possible soit un aléatoire indéterministe, même si cela ne signifie rien d'autre que supposer l'existence d'aléatoire indéterministe. Quand la théorie de la mécanique quantique fut lancée dans les travaux de Planck, Einstein et Bohr, il était clair que la physique classique ne pouvait pas expliquer des phénomènes à si petites échelles, tels que l'interaction entre particules. La théorie de la mécanique quantique le pouvait, mais à grand coût, car il fallait être prêt à faire toutes sortes de concessions au sujet de certains phénomènes incompatibles avec la physique classique. Pour Bohr, ces concessions constituaient une réalité au niveau du monde quantique, tandis qu'Einstein pensait que la bizarrerie de la mécanique quantique signifiait que la théorie était incomplète, qu'à la fin cela ne pouvait pas être aussi incompatible que ça l'était par rapport à la réalité familière que nous habitons. Il était convaincu qu'une théorie encore à découvrir sur des variables cachées pourrait expliquer certains, sinon tous ces étranges phénomènes.

La conclusion logique qu'on peut tirer de la violation du réalisme local, c'est qu'au moins une de ses suppositions est fausse. Mais on pourrait considérer que l'échec d'autres suppositions implicites dans notre raisonnement conduise à l'inégalité. Il pourrait y avoir parmi elles une violation de la limite imposée par la théorie relativiste ou bien un monde qui soit complètement déterministe [Bell, 1985].

On dit souvent que les physiciens quantiques s'occupant de réalité quantique évitent de traiter cet échappatoire, en supposant qu'on ne peut réaliser aucun test sur le déterminisme complet. Néanmoins, le pseudo-aléatoire a

ceci de particulier que lorsque son processus de génération est répété, le résultat est exactement le même. Dans ce cas, le même phénomène va aussi se produire en mécanique quantique. La difficulté vient peut-être du fait que le formalisme de la mécanique quantique ne nous donne que des amplitudes de probabilités.

3.5 Vers une théorie informatique de la physique

D'un autre côté, on pose le caractère fondamental de l'information dans la physique moderne, les trous noirs, la théorie des cordes, etc. La supposition que l'information est essentielle pour expliquer notre réalité physique n'est pas étrangère à la physique moderne. Beaucoup de physiciens sont arrivés aux mêmes conclusions, en donnant au concept de symétrie (qui peut se voir comme une information ou comme un objet mathématique abstrait) un rôle de base, jusqu'à lui faire prédire l'existence de nouvelles particules, par exemple. Et jusqu'ici, cette approche a été une réussite dans de nombreux domaines de la physique. Il est remarquable de voir comment une simple description "informationnelle" de certaines propriétés fondamentales de la mécanique quantique décrit pleinement le phénomène quantique. De telles descriptions spécifient comment les particules interagissent et comment elles sont reliées entre elles par des symétries.

Selon la mécanique classique, l'aléatoire n'est qu'apparent dans le monde macroscopique, mais selon l'interprétation standard de la mécanique quantique, il en va tout autrement, et de façon fondamentale. La position selon laquelle l'histoire du monde est computationnellement réversible est compatible avec le déterminisme imposé par la mécanique classique. Cependant, à l'échelle quantique, les choses semblent différentes. Quand on intrique une particule, celle-ci semble véritablement et irréversiblement perdre trace de son état antérieur, et il n'y a aucun moyen, même en inversant toutes les opérations, de le retrouver. Il se passe la même chose avec la désintégration radioactive, où le noyau atomique d'un atome instable perd de l'énergie en émettant des particules ionisées, dont il est impossible, selon la théorie quantique, de prévoir le moment de désintégration, celle-ci pouvant arriver à tout moment, de façon équiprobable.

Zeilinger prétend que l'aléatoire quantique est intrinsèquement indéterministe, et que les expériences violant les inégalités de Bell impliquent que certaines propriétés n'existent pas tant qu'elles ne sont pas mesurées. Tou-

tefois, ces déclarations se basent sur une interprétation (Copenhague) particulière de la de la mécanique quantique (voire une spéculation sur celle-ci), à partir de laquelle il bondit jusqu’aux conclusions en s’appuyant sur divers théorèmes anti ou sans-variables-cachées, proposés par des gens comme von Neumann, Bell, Kochen et Specker, qui sont supposés démontrer que l’aléatoire quantique est vraiment indéterministe. La position de Zeilinger est une forme d’aléatoire épistémologique faible, lorsqu’il soutient qu’on ne peut observer moins d’un bit d’information, et que ce qui n’est pas connu est alors “aléatoire”.

Ma position est en résonance avec la réponse à Zeilinger dans [Daumer et al., 2006]. Mais bien qu’on partage avec Daumer et al. la conviction que Wheeler n’apporte pas beaucoup d’éclaircissement au problème avec le captivant mais néanmoins assez obscur traitement qu’il réserve à l’information, de la façon dont il l’associe ou la positionne comme plus fondamentale que la physique (son dicton “it from bit”), on ne partage pas leurs vues sur ce qui est faux quant au contenu informationnel de la mécanique quantique. Comme ils le soulignent, la remarquable suggestion de Wheeler était que la physique ne porte que sur de l’information ou que le monde physique lui-même est information. Selon les mots de J. A. Wheeler lui-même [Wheeler, 1990] :

Il n’est pas déraisonnable d’imaginer que l’information siège au cœur de la physique, tout comme elle siège au cœur d’un ordinateur. *It from bit*. Posé autrement, tout *it*—toutes particules, tous champs de force, même le continuum espace-temps—dérivent entièrement leurs fonctions, leur signification, leur existence même—parfois indirectement dans certains contextes—de réponses, arrachées à un appareil, à des questions oui/non, à des choix binaires, bref de bits.

Wheeler, cependant, a introduit une nuance liée au sens, lorsqu’on fait entrer en scène l’observateur (il faut rappeler que Wheeler a aussi à son crédit la formulation du principe anthropique). Ainsi, dans la vision de Wheeler, le sens a aussi un côté subjectif, ce qui veut dire que ce n’est pas une vision complètement réductionniste et triviale du contenu informatif.

“It from bit” symbolise l’idée que tout item du monde physique possède un arrière-plan—très profond, dans la plupart des cas—une source et une explication immatérielle ; et que ce qu’on appelle réalité survient, en dernière analyse, des questionnements oui/non et de l’enregistrement de réponses suscitées par des équipements ; en bref, que toute chose physique est de l’information théorique à l’origine et que c’est un univers participatif.

Ce que suggère Wheeler de plus pressant, c’est que “l’information” ne peut se définir en terme de “matière” ou “d’énergie”, les notions les plus basiques en physique. Et un second niveau de lecture introduit le problème du contenu informatif, du sens, et de l’observateur dans son univers participatif.

3.6 Une approche algorithmique du problème de l’ajustement précis (fine-tuning)

Le principe anthropique (les paramètres fondamentaux de l’univers doivent être tels qu’ils permettent la naissance d’observateurs) peut avoir une interprétation algorithmique fournissant une réponse (plus raisonnable) à la question de l’ajustement précis de l’univers, en gros la question de savoir pourquoi l’univers paraît exactement comme il faut pour que tout s’arrange pour donner naissance à des observateurs. Un seul changement de valeur dans l’équation ou un seul des paramètres produirait un univers où rien ne serait possible, et certainement pas la vie. Mais si l’univers est un programme d’ordinateur, les paramètres ne se contentent pas d’être couplés, mais il peut y avoir de nombreux programmes fournissant le même résultat, surtout un résultat de basse complexité algorithmique, ce qui en fait un univers structuré plutôt qu’apparemment aléatoire—en opposition à Calude et al. [[Zenil, 2012a](#)]. Le fait que les paramètres soient couplés est aussi un reflet d’une possible *ToE* (ou Théorie du Tout), selon laquelle tout vient à exister en tant que résultat d’une seule loi physique. D’où on peut s’enquérir d’autres possibles Théories du Tout, qui modéliseraient bien le comportement d’univers où les choses semblent s’ajuster finement, comme elles le font dans le nôtre. Dans l’approche de Wolfram, par exemple, les univers informatiques qui ne deviennent pas très vite triviaux, commencent à nous sembler assez compliqués pour juger avec évidence si c’est ou non notre propre univers qui est modélisé. Cela signifie qu’il existe un seuil (saisi dans le Principe d’Equivalence Computationnelle de Wolfram) au-delà duquel un univers peut commencer à avoir l’air finement ajusté pour accommoder toute la structure dont il est capable, sans être du tout un univers spécial. Au contraire, comme il est capable de structure, il doit avoir une complexité algorithmique basse et donc une haute probabilité algorithmique.

En suivant Wheeler, je pense que le prochain niveau d’unification (au fil de l’histoire des unifications d’autres concepts auparavant séparés en science, comme électricité et magnétisme, lumière et électromagnétisme, énergie et

masse, pour en citer quelques-unes) impliquera l'information et la physique (et finalement, comme une conséquence, la computation et la physique).

Landauer a écrit en 1996 (cité d'après Leff and Rex, pag. 335 [[Leff and Rex, 2002](#)]) :

L'information n'est pas une entité abstraite désincarnée ; elle est toujours liée à une représentation physique. Elle est représentée par une gravure sur une tablette de pierre, un spin, une charge, un trou percé dans une carte, une marque sur du papier, ou autres équivalents. Cela relie le traitement de l'information à toutes les possibilités et restrictions de notre monde physique réel, à ses lois physiques et à la gestion de ses parties disponibles.

Les tentatives actuelles pour unifier la physique moderne, telles que la théorie des cordes et aussi la gravité quantique (où dans certaines versions de gravité quantique c'est l'information qui est l'élément fondamental), dépendent de l'encodage de l'information et de combien d'information on a besoin pour définir quelque chose. Dans ces modèles, c'est l'information qui donne sens aux forces de l'univers et à la matière elle-même. Dans la gravité quantique, espace et temps ne sont pas fondamentaux ; c'est l'information qui constitue le niveau le plus basique de la réalité physique. Voilà, tout vient de l'information. La relativité avait déjà relégué l'espace et le temps dans ce statut, car il n'y a rien de spécial dans l'une ou l'autre dimension ; c'est juste ce qui se passe dans chacune qui différencie l'espace du temps. Les événements qui y prennent place n'ont qu'une signification subjective, avec l'information jouant un rôle fondamental. En gravité quantique aussi, l'échange d'information est fondamental. Ces théories sont supposées introduire une compatibilité entre mécanique quantique et relativité générale grâce au concept d'information (en particulier, l'information maximum), en connectant l'énergie de la théorie quantique et l'énergie de la théorie de la relativité, pour finalement bâtir un pont entre les deux.

3.7 Les trous noirs en tant que possibles compresseurs avec pertes de données

Leo Szilárd [[Szilard, 1929](#)] et Rolf Landauer [[Landauer, 1961](#)] ont déterminé la façon dont un ordinateur utilise l'énergie lorsqu'il traite de l'information. Il se trouve qu'on peut voir les ordinateurs dépenser de l'information uniquement quand ils en effacent. On peut ajouter, multiplier ou soustraire

des bits sans consommer ou augmenter l'énergie de l'univers, tant qu'aucune information n'est effacée lors du processus. Le principe de Landauer vient de principes physiques bien établis. Malheureusement, on doit presque toujours effacer de l'information, pour presque tout. L'effacement libère de la chaleur, qui se dissipe dans l'environnement. On en déduit que l'effacement n'est pas une opération réversible. Effacer n'est pas comme soustraire ou ajouter, car soustraction et addition sont des opérations réversibles, tandis que l'effacement entraîne une perte d'information locale, la chaleur libérée contenant l'information perdue. La perte d'information au niveau de la mécanique quantique est apparemment d'une nature différente.

Il s'avère que les trous noirs jouent un rôle important dans la vision informationnelle du monde, étant donné que l'information maximum, quand les bits correspondent à des photons, est la masse maximum déterminée par la limite du trou noir. Au départ, on pensait que les trous noirs détruisaient l'information de l'univers ; tout ce qui tombait en eux devait être irréversiblement détruit, malgré la supposition généralisée, tirée de la thermodynamique, que l'information dans l'univers n'est jamais détruite [Bennett, 1982]. Cependant, on s'est aperçu plus tard que même les trous noirs pouvaient conserver l'information sous la forme d'un phénomène quantique apparaissant à l'horizon événementiel, en émettant de petites quantités de radiations thermiques [Hawking, 1974], à partir desquelles toute information tombée dans le trou noir pouvait être entièrement retrouvée, en principe.

En fait, on peut maintenant déterminer la formation des trous noirs, dans le contexte de ce qu'on a appelé le principe holographique [Susskind, 1995], donnée par le nombre de bits d'information égal à la surface du trou noir divisé par quatre et multiplié par la surface de Planck ($10^{-66}m^2$). Parmi plusieurs conséquences importantes, il y a que l'information maximum concernant cet objet physique peut se comprendre en termes d'information en deux dimensions (sa surface), raison pour laquelle le principe associé est appelé holographique—par analogie avec le fait que la surface bidimensionnelle d'un objet contient de l'information sur son volume en trois dimensions [Bekenstein, 1980]. Cette thermodynamique des trous noirs définit la quantité maximum d'information qui peut se stocker dans une région donnée finie de l'espace possédant une quantité finie d'énergie. Plus un trou noir recèle d'information, plus son horizon événementiel est grand, et aucune place plus petite ne pourrait en contenir autant, les trous noirs ayant leur taille maximum quand ils atteignent le point de singularité. Les trous noirs peuvent donc se voir comme de compresseurs imparfaits de données (avec pertes à cause du rayonnement entropique de Hawking).

La décohérence quantique signifie que l'information possédée par un ob-

servateur, au sujet des probabilités des différents résultats possibles, nécessite certains degrés physiques de liberté spécifiques. Mais encore, le problème de la mesure repose seulement sur le fait que l'observateur ne connaît pas le résultat d'une mesure quantique, et que ces mesures quantiques paraissent ne pas être réversibles, car re-intriquer une particule pour réaliser la même expérience conduirait à un résultat différent, résultat divergeant de la mesure précédente non parce qu'il est aléatoire au sens mathématique (algorithmique), mais dans le même sens que quand on ne sait pas, quand nous nous sentons incapables de prévoir le monde classique, non parce qu'il est indéterministe, mais parce qu'il est imprévisible. Même s'il y a de subtiles différences, tout ceci revient à un problème de nature épistémologique, avec l'information au bout. Cela conduit à montrer comment l'information peut déjà expliquer certains des processus les plus fondamentaux dans l'univers, au plus près de notre compréhension du plus grand et du plus petit, et souligne la pertinence de se poser sérieusement la question de savoir si l'univers peut pleinement s'expliquer en termes d'information et computation, peut-être même suggérer des moyens de concilier la relativité générale avec la mécanique quantique.

3.8 Cartographie simplifiée des visions (in)calculables du monde

Aujourd'hui information et computation jouent un rôle majeur en physique moderne, étant toutes deux sources de nouvelles théories unificatrices, et permettent aussi pour des approches judicieuses de certains aspects des théories dominantes comme la mécanique statistique et la thermodynamique, où on a prouvé leur grande utilité. Elles sont effectivement essentielles pour de nombreux concepts physiques, de nos jours.

Concerné comme j'étais par toutes ces questions, ce fut un privilège d'avoir l'occasion de co-organiser et d'être impliqué dans plusieurs événements autour de ce sujet. En 2008, avec Adrian German et Gerardo Ortiz, j'organisai la seconde *Midwest NKS Conference* à l'université d'Indiana Bloomington, comprenant un ensemble impressionnant de participants et d'intervenants (voir <http://www.cs.indiana.edu/~dgerman/2008midwestNKSconference/>) dont Bennett, Calude, Chaitin, Csicsery, Deutsch, Fredkin, Grover, Leggett, Lloyd, Rowland, de Ruyter, Szudzik, Toffoli et Wolfram. L'élan suscité par la conférence contribua à la réalisation de ce projet, dont la transcription de la contribution de Deutsch à cet ouvrage,

et la transcription de la table ronde sur le sujet, incluse aussi dans la présente thèse. En 2010, avec Tommaso Bolognesi, le *JOUAL* 2009 Workshop a été organisé (voir <http://fmt.isti.cnr.it/JOUAL2009/>), portant sur des questions autour des concepts d'émergence, d'espace-temps et de nature dans les systèmes computationnels. Étaient présents Loll, Wolfram et Schmidhuber, parmi d'autres orateurs.

Une importante question à laquelle cette thèse peut suggérer une réponse, c'est si ces visions sont assez matures pour être prises en compte et étudiées de long en large. J'ai eu le privilège de piloter l'effort pour relever ce défi pour un recueil [Zenil, 2012c], et en définitive, on pense qu'on a eu un livre compréhensible, dans lequel la plupart, sinon toutes les tendances actuelles sont présentes, d'une façon ou une autre. En préparant cette thèse, j'ai eu à m'assurer que des opinions contraires étaient bien représentées, en l'occurrence les points de vue de ceux qui ne sont pas d'accord avec la thèse principale sur un univers (Turing) calculable, notamment des opinions venant du champ important de la mécanique quantique (en commençant par Penrose, qui a écrit la préface de l'un de mes livres (A Computable Universe [Zenil, 2012a]), en incluant aussi Zizzi, Lloyd, Deutsch et Cabello). Présents aussi sont ceux qui épousent, par un biais ou un autre, une certaine idée d'hypercomputation selon telle ou telle terminologie (Doria, Cooper, Penrose et Zizzi encore), et des penseurs représentant une nouvelle tendance, partisans d'un monde algorithmiquement aléatoire (Calude, Meyerstein, Salomaa et Svozil), tendance qui se trouve être diamétralement opposée à ma propre vision algorithmique. En m'efforçant de fournir une cartographie pratique pour se déplacer parmi ces points de vue, je les ai groupés en quelques catégories, pleinement conscient que je prend le risque d'une sursimplification. Certaines de ces catégories s'opposent, ou sont des ramifications d'une catégorie plus large : par exemple digital versus quantique, déterministe versus aléatoire. Ce n'est que ma contribution personnelle simplifiée, qui ne représente nullement nécessairement ce que les auteurs auraient défendu, que ce soit pour leur travaux ou les travaux des autres. C'est simplement dans l'intention d'aider le lecteur dans la comparaison des points de vue, et qu'il ou elle s'oriente librement, face à plusieurs hypothèses.

- L'Hypothèse de l'Univers Computable (Wolfram, Bolognesi, Szudzik, Schmidhuber, Hutter). D'après moi, leur point de vue est surtout épistémologique par nature, contrairement à la vision leur attribuant une position ontologique, bien qu'il puisse y avoir plusieurs degrés depuis la position purement ontologique (par exemple Fredkin, héritier de la sous-catégorie) jusqu'à la position épistémologique (par exemple le formalisme de Szudzik [Zenil, 2012a] qui saisit le contenu épistémo-

logique de cette hypothèse, tandis que Bolognesi [Zenil, 2012a] vise à fournir une base ontologique d'où émergerait la physique). Tomberaient dans cette catégorie des positions comme celles de Wheeler et peut-être Feynman. La plupart des contributions sur la gravité quantique pourrait aussi tomber dans cette catégorie (même si les auteurs pourraient le contester ou se demander si c'est le cas).

- L'Hypothèse de l'Automate Cellulaire. Une sous-catégorie suggérée d'abord par Zuse, puis adoptée et développée par Fredkin selon son programme de *Philosophie et Physique Digitale* [Zenil, 2012a]. Traditionnellement peu soutenue mais très apte à fournir des concepts fondamentaux pour le sous-champ de la physique de la computation (par exemple des questions relatives à la réversibilité logique et physique).
- L'Hypothèse de la Structure Mathématique. Une sous-catégorie de l'Hypothèse Computationnelle. Suggérée par Max Tegmark [Tegmark, 2007] et couverte par l'hypothèse de l'Univers Computable, étant donné que Tegmark fait allusion à une structure mathématique qu'il considère computable (des versions non-computables peuvent se regrouper sous l'hypothèse de l'Univers Non-Turing Computable).
- L'Hypothèse Algorithmique Epistémologique. Les modèles appartenant à cette catégorie sont surtout agnostiques par rapport à toute déclaration ontologique concernant la structure ultime du monde. Ils sont pragmatiques dans leur approche des phénomènes naturels et des applications concrètes basées sur un fondement informationnel ou computationnel (par exemple [Zenil, 2012a] Ehrenfeucht et al., Rozenberg, Martinez, Adamatzky, Teuscher, Vellupillai et Zambelli). Ses valeurs sont indépendantes d'une vision du monde particulière.
- L'Hypothèse Algorithmique (proposée dans cette thèse) opposée à l'hypothèse Aléatoire. Elle prend l'Hypothèse Algorithmique Epistémologique à un niveau ontologique en tant qu'explication de la plupart, sinon de toutes les structures dans le monde, grâce à la probabilité algorithmique (et donc la computation) décrivant la distribution des patterns dans la nature. Elle peut remplacer le niveau ci-dessus mais n'est pas conçue pour décrire les blocs ultimes de construction de l'univers (ou sa puissance computationnelle) contrairement à d'autres niveaux cités plus haut, mais plutôt pour expliquer la structure comme le résultat de processus algorithmiques (computationnels).
- L'Hypothèse Non-Turing Calculable/Computable (vision représentée par Doria [Zenil, 2012a], et de façons différentes, avec des étiquettes ou

des approches différentes, par Calude et Cooper), en opposition avec la théorie de Church-Turing dans ses diverses versions. Que ce soit en acceptant le point de vue que la nature peut encoder (et résoudre) le problème de l'arrêt et donc contenir en elle de la computation de 'type 2' (ou hypercomputation), ou bien en croyant à l'existence de régularités incalculables, ce qui revient à dire que, d'une certaine façon, on peut qualifier de régulière une irrégularité (c.-à-d. un "pattern" non calculable), ce qui suppose qu'il existe quelque chose dans la nature (par exemple le cerveau) qui est capable de trouver des patterns là où, même en principe (même dans un temps illimité), les ordinateurs ne peuvent le faire (par exemple "voir" un Oméga de Chaitin sans la machine qui l'a fabriqué, alors que nous ne sommes même pas capables de reconnaître des patterns/motifs dans des nombres calculables comme π). Parmi d'autres partisans importants de cette hypothèse, on peut citer des chercheurs comme J. Félix Costa, Mark Burgin et Selmer Bringsjord.

- L'Hypothèse Aléatoire (Calude, Meyerstein, Salomaa, McAllister). Peut se voir comme une sous-branche de l'hyper-computation, mais avec des arguments différents et une nouvelle approche qui utilise la théorie de l'aléatoire algorithmique, tout en embrassant potentiellement l'interprétation forte (en l'occurrence l'interprétation dominante) de la mécanique quantique (celle de Copenhague).
- L'Hypothèse de l'Univers non-Computable (Penrose). Des arguments pouvant appartenir à l'hypothèse de l'Hyper-computation et à l'Hypothèse Quantique. D'autres partisans possibles (avec des variations) sont Lucas et Searle. Opposée à la plupart des autres points de vue, mais peut-être tangentielle avec l'Hypothèse de l'Hyper-computation.
- L'Hypothèse Quantique (Lloyd, Deutsch, Cabello, Zizzi, Penrose) : Physique et complexité basées sur la mécanique quantique.

Chapitre 4

Vers une vision algorithmique du monde

La discussion sur la nature digitale ou analogique du monde n'a pas de rapport avec l'approche présentée ici. L'approche algorithmique présentée ici peut laisser supposer que le monde est digital (pour des raisons expliquées plus tard) mais elle peut être compatible avec des conceptions autres que le digital (même si le digital suffirait à justifier ma position sur le comportement algorithmique du monde). Nous allons cependant discuter l'idée que c'est la façon dont le monde se déploie qui est importante, d'où on ne peut pas tirer la conclusion que le monde est digital mais d'où on peut déduire qu'il n'a pas l'air aléatoire.

Même si la science a parié sur la conception d'un monde causal et basé sur des règles, concevoir le monde comme algorithmique entraîne d'importantes conséquences, particulièrement parce qu'on peut y appliquer des outils tirés du sous-domaine informatique de la complexité algorithmique, à la lumière des développements scientifiques actuels et des conceptions de notre réalité physique.

Il est légitime de se demander pourquoi le monde semble souvent si organisé. On voit des planètes orbitant autour d'étoiles, des étoiles se groupant en galaxies, des cristaux se formant et grandissant en surfaces, et même de la matière parvenant à la vie, développant l'intelligence et la conscience d'elle-même.

Parmi les plus simples hypothèses compatibles avec le meilleur compte-rendu disponible de l'origine de l'univers, on choisit celle qui démarre en basse complexité – l'état de l'univers avec toute sa matière et son énergie ramassées en un point infiniment petit sans longueur ni largeur, et d'une

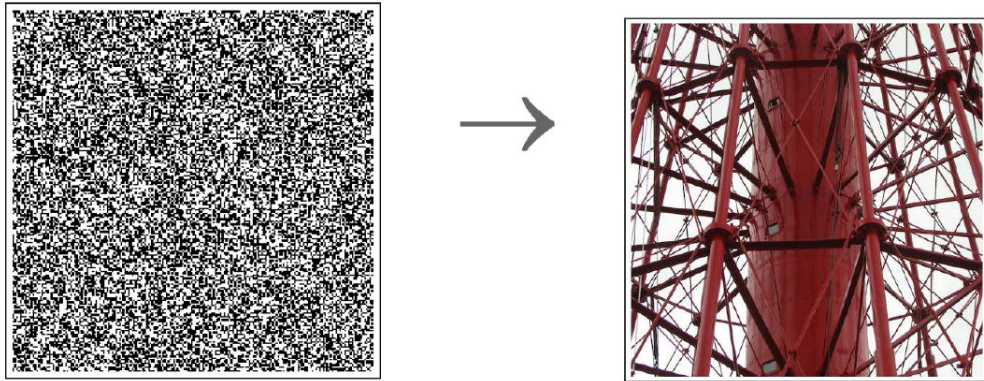


FIGURE 4.1 – Du bruit à la structure. La radiation thermique de fond (à gauche) – ou l’apparence de l’univers à toutes les échelles dans toutes les directions – et les types de structures (à droite) que nous rencontrons aujourd’hui dans la vie de tous les jours.

densité infinie. Il y a dû y avoir une transition vers l’état où nous nous trouvons aujourd’hui, dans un univers aux lois physiques gouvernant tous les aspects de notre réalité, des plus grands aux plus petits, des lois souvent assez simples pour être facilement compréhensibles, même si les phénomènes qu’elles décrivent sont compliqués.

L’univers semble hautement ordonné et structuré aujourd’hui, comparé au bruit de fond laissé après le *Big Bang*, semblable à ce qu’on peut voir sur l’écran d’une vieille télé analogique mal réglée (Fig 4.1) ou à ce qu’on peut entendre sur un poste de radio déréglé. Étant donné l’état aléatoire que les physiciens supposent avoir existé au tout début, incorporant peu ou pas du tout d’information, comment en sommes-nous arrivés au monde structuré dans lequel nous vivons ?

Que l’univers ait commencé son existence sous la forme d’un point isolé, ou qu’il ait commencé par un état de complet aléatoire, à la fois le point et l’état d’aléatoire peuvent se voir comme des quintessences de symétrie parfaite – car aucune partie ne possédait plus ou moins d’information qu’une autre, car il n’y avait aucune partie, ou parce qu’aucune partie ne transportait l’information, comme le bruit de l’écran de télé mal réglée. L’équilibre thermique dans un espace en expansion était instable et des asymétries commencent à apparaître. Certaines régions devinrent plus froides que d’autres et commencèrent à produire les premières structures.

Quand l’univers s’est refroidi au point où les atomes les plus simples purent se former, le bruit blanc a cessé de dominer et la matière décolla,

embarquée dans un processus de formation de structure. La brisure de symétrie pouvait se voir partout dans l'univers, de la grande disparité entre matière et antimatière (des atomes avec particules à charges inversées) jusqu'à la manière dont les planètes tournoyaient dans un seul sens autour du soleil ; en passant par la forme de ce qu'on connaît aujourd'hui sous le nom d'homochiralité (des groupes de molécules dans lesquels il manque une des configurations en miroir, non superposables) ; et en ce qui concerne les êtres vivants, tous ceux qui partagent les mêmes aminoacides et sucres, mais génétiquement codés de telle manière que chacun possède une orientation moléculaire (arbitraire) plutôt qu'une autre. Les lois de la physique peuvent avoir surgi de cette façon, non comme des agents façonnant l'univers mais comme le résultat du déploiement de cette dynamique de traitement de l'information sous forme de création de structure.

Je vais développer ce que je pense être les propriétés algorithmiques de l'univers, en fonction de la façon dont il se déploie, c'est-à-dire les motifs, les structures et le type d'aléatoire qu'on obtient dans notre monde. Se pourrait-il que la complexité qui caractérise l'univers physique n'émerge que de l'itération de quelques simples règles (algorithmiques) ?

Ce serait trop facile de se contenter de dire que de hauts niveaux d'organisation peuvent surgir d'un état désordonné, comme beaucoup le font sans fournir aucune preuve ou support théorique d'une telle affirmation. Je vais apporter ce que je pense être un cadre théorique plausible. Et nous avons aussi entrepris un travail empirique [Delahaye and Zenil, 2012] pour rechercher les données à l'appui de ce que je vais débattre ici, sur lequel nous basons notre affirmation qu'il se trouve en effet qu'à partir d'aléatoire on finisse par obtenir non seulement une structure organisée mais une structure qui pourrait bien ressembler aux distributions tirées de données empiriques. Les structures se révèlent aussi assez stables et robustes au sens où elles ressemblent aux distributions issues de divers modèles algorithmiques, et pas d'un seulement.

L'univers s'auto-alimente avec les résultats de ses états précédents, générant pour lui-même certaines entrées d'allure aléatoire et produisant, à haute fréquence, des motifs (structures) à l'air non-aléatoire comme la théorie le prévoyait, générant les entrées pour le prochain calcul (bien qu'il puisse être séquentiel, il se peut qu'il n'ait pas besoin d'être synchronisé, car certains événements peuvent venir d'une série différente d'événements précédents).

Quel que soit l'état dans lequel il était quand le Big Bang s'est produit, c'était alors sa plus courte description – si description il y avait – et l'univers pourrait bien être le résultat des règles simples algorithmiques. Wol-

fram pense que dans l'énumération naturelle des possibles candidats-univers computationnels, le nôtre apparaîtrait tôt dans la liste, car plus la théorie est simple, plus elle a de chances de s'accorder avec nôtres contraintes physiques, par exemple, ceci conduisant à la complexité présente autour de nous. Cette vision est compatible avec une interprétation de la probabilité algorithmique selon laquelle un univers structuré est le résultat d'un processus et aura donc tendance à apparaître tôt dans une énumération des états initiaux possibles, seulement parce que plus il est structuré, plus grandes seront les chances qu'il parvienne à exister.

4.1 La structure née de l'aléatoire

C'est une idée fausse assez courante de croire que les programmes informatiques ne peuvent jamais produire quelque chose allant au-delà de ce que le programmeur avait décidé. Et quand on ne pose pas de contraintes à la base, certains programmes ont tendance à s'emballer, donnant toutes sortes de comportements, comme Wolfram l'a montré [Wolfram, 2002]. Wolfram a montré que même les plus simples programmes faisaient preuve de comportements inattendus et compliqués. On l'avait remarqué auparavant, sous la forme d'un phénomène que les Grecs décrivaient en terme d'irrationalité – le fait que certaines grandeurs, lorsque divisés par d'autres, provoquent un calcul infini sans régularité apparente qui permettrait de le décrire complètement d'une façon condensée, comme on peut le faire avec le résultat du quotient de deux entiers. Lors de l'investigation systématique de Wolfram, celui-ci trouva que même les plus simples programmes informatiques se comportent souvent de manière imprévisible, imprédictible au sens où on se retrouve incapable d'expliquer le déploiement du système, et incapable de trouver des raccourcis pour obtenir le résultat sauf pour gagner quelques étapes de calcul, ou alors en accélérant le calcul, mais pas moyen de shunter le nombres d'étapes.

Pour produire les décimales de la constante mathématique π en tirant des digits au hasard, il faudrait essayer encore et encore jusqu'à obtenir des chiffres consécutifs concordant avec un segment initial de l'expansion des décimales de π . La probabilité de succès serait infime, dépendant de la longueur du segment désiré collant avec π , ce qui donne $1/10$ multiplié par le nombre de digits désirés. Par exemple, $(1/10)^{2400}$ pour un segment de seulement 2400 digits de π . Mais si au lieu de tirer les digits au hasard, on tirait des bits de programmes informatiques et on les exécutait sur un ordinateur, les choses changeraient du tout au tout. Par exemple, un programme donnant les digits de la constante mathématique π aurait plus de chance d'être produit par

un programme informatique. Voici un exemple de programme écrit en ANSI C, un langage de seulement 158 caractères, qui code une formule pour π et donne ses 2400 premiers digits.

```
int a = 10000, b, c = 8400, d, e, f[8401], g;
main()
{
    for(;b-c;) f[b++] = a/5;
    for(; d = 0, g = c * 2; c -= 14,
        printf(''\%.4d'', e + d/a), e = d \% a)
        for(b = c; d += f[b] * a, f[b]= d \% -g, d /= g-, -b; d *= b);
}
```

Quand on spécifie la longueur des programmes, il peut sembler nécessaire d'ajouter la taille du compilateur en langage C à la taille du programme. Le programme peut avoir l'air de dépasser son réel degré de compression algorithmique car les 158 caractères de C ne produisent pas les 2400 digits sans qu'ils soient aussi combinés avec un compilateur C, qui contient lui-même une information significative. Ainsi, jeter le dé en l'air ne devrait pas avoir seulement comme résultat le programme C lui-même mais aussi le compilateur, pour interpréter correctement et exécuter le programme.

Quand on compare les programmes informatiques, on doit le faire sur la base d'un langage commun. Si le langage commun est C, comme ici, on peut ignorer la taille du compilateur car c'est la même pour tous les programmes. En d'autres mots, parce que la constante additionnelle est commune à tous les programmes, on peut l'ignorer. Le théorème d'invariance montre que ce n'est pas très grave si on ajoute ou non la longueur du compilateur, ni très important le choix du langage utilisé, car entre deux langages informatiques L et L' , il existe une constante c qui dépend seulement des langages informatiques et non de la chaîne, de sorte que pour toutes les chaînes s : $|K_L(s) - K_{L'}(s)| < c_{L,L'}$, où $c_{L,L'}$ est une constante c qui ne dépend que des langages L et L' et non pas de la chaîne s . Il faut avoir à l'esprit que cela équivaut à dire qu'il y a toujours un traducteur de longueur fixe (un autre compilateur entre langages) dont on peut se servir pour décrire la longueur des programmes, sans avoir à trop se soucier des constantes additionnelles et sans perte de généralité.

Il existe beaucoup d'autres formules qui peuvent s'implémenter en tant que programmes courts dans n'importe quel langage pour générer un nombre arbitraire quelconque de digits de π . Il est souvent possible d'écrire des sous-routines pour raccourcir un calcul. Dans Mathematica par exemple, on peut

obtenir n'importe quel digit de π simplement en exécutant $N[\pi, n]$, avec n le nombre de digits désirés. Finalement, la définition de la complexité algorithmique d'une chaîne est donnée en terme de longueur de bits du plus court programme produisant la dite chaîne. Ainsi le choix du langage de programmation est immatériel, car tous les langages sont comparables quand ils sont traduits en code machine, c'est-à-dire que tous sont écrits dans un langage commun (binaire).

Le langage C particulier calculant les premiers 2400 digits de π ne se sert d'aucune fonction particulière de C, juste d'opérations mathématiques de base.

4.2 Complexité algorithmique dans le monde réel

On s'accorde à dire que la complexité algorithmique est le cadre approprié pour discuter de l'opposition simplicité/complexité en science. Cela vient du concept de la plus courte description possible d'un objet. L'idée est que si le plus court programme pour produire une chaîne, est par exemple à peu près de la même longueur que la chaîne, alors cette chaîne est complexe ou aléatoire, tandis que si le programme est beaucoup plus court que la chaîne d'origine, alors cette chaîne est compressible, donc elle est simple, et si elle ne l'est pas alors elle est aléatoire.

L'information a évolué à partir d'une situation primitive (quand l'univers était si dense que tout semblait aléatoire) jusqu'aux formes actuelles, plus organisées, qu'on voit aujourd'hui (par ex. il est presque certain que dans l'univers primitif il n'y avait pas de vie, car il semblerait que la vie requiert une longue période de calcul avant d'émerger).

La complexité algorithmique de l'univers primitif devait être très basse, entièrement compressé comme il l'était, mais on peut supposer qu'il est resté constant jusqu'à aujourd'hui et qu'il est en principe possible de le recompresser maintenant comme il était jadis, au début. Cependant, si on prend la profondeur logique de Bennett comme une mesure de la complexité algorithmique, on peut alors en déduire que la complexité du monde peut parfois augmenter, car elle est basée sur le concept de temps de décompression à partir de l'origine compressée – et cette idée nous séduit en tant que belle description de l'évolution de l'univers.

Selon la profondeur logique de Bennett, la complexité d'un objet est donné par le temps de décompression d'un programme proche du plus court possible

produisant l'objet. Cette mesure possède la particularité de faire la distinction entre ce qui est aléatoire et ce qui est structuré et organisé, contrairement à la pure complexité algorithmique.

Malheureusement, de telles mesures sont encore largement sous-utilisées, et souvent négligées ou même mal comprises. Le gros problème est qu'elles ne sont pas calculables, cela veut dire que lorsqu'on est en présence d'une chaîne, il n'y a pas d'algorithme qui nous donne une quelconque valeur de complexité. Pour cette raison, on a soigneusement évité d'essayer quoi que ce soit avec elles. Toutefois, il y a eu des tentatives pour bâtir des outils basés sur ces concepts, et c'était une partie de notre programme de recherche [Delahaye and Zenil, 2012, Joosten et al., 2011, Delahaye and Zenil, 2007]. Selon la profondeur logique de Bennett [Bennett, 1988], les structures dans l'univers évoluent depuis 14 milliards d'années (le temps mis par l'univers pour s'étendre depuis le Big Bang), elles ont eu l'opportunité de devenir plus complexes lors de leur déroulement historique (s'il n'est pas trivial). Si le début de l'univers a l'air aléatoire mais conduit à la complexité, cela voudrait-il dire que cet état aléatoire est hautement compressé ? Que nous dit la cosmologie sur la compressibilité de l'univers ?

La question est trompeuse car elle semble se répondre à elle-même, si on suppose une expansion de l'univers basée sur la complexité algorithmique. Mais en fait, la réponse a deux facettes. L'univers n'est pas seulement compressé au tout début, lorsque le Big Bang s'est établi. Si l'univers était compressé, alors c'est parce qu'il était compressible. Par conséquent, non seulement il n'est pas aléatoire, mais il est ouvert à une complète description algorithmique, aucune information supplémentaire ne fut introduite plus tard, et les choses sont restées causalement connectées. Mais cela signifie aussi qu'il n'y a rien d'autre que du pseudo-aléatoire dans l'univers, simplement parce que si rien n'y a été introduit comme nouvelle information, alors l'univers s'est déplié à partir d'un état primitif considéré comme sa description compressible.

4.3 La place de la mécanique classique

Certains phénomènes peuvent être modélisés en supposant que matière et espace existent comme un continuum, signifiant que la matière est distribuée de façon continue dans une région entière de l'espace. Par définition, un continuum est un corps pouvant être continuellement sous-divisé en éléments infinitésimaux. Cependant, la matière est constituée de molécules et d'atomes

séparés par un espace vide. Si un modèle comme la relativité générale est accepté comme description du monde à toutes les échelles, alors on devrait aussi considérer la matière comme un continuum.

Les modèles discrets paraissent capables de saisir toutes les symétries-clé requises de la part d'une théorie physique compréhensible. Les modèles discrets sous forme de formules et de modèles généraux ont, eux aussi, réussi assez bien à modéliser notre monde. Certains d'entre eux peuvent sembler essentiellement discrets, mais ce n'est qu'en apparence car pour résoudre une équation, (en supposant que des solutions existent), on doit avoir un algorithme faisant les calculs suggérés par l'équation, ce qui les rend algorithmiques, sinon juste digitaux. Wolfram [[Wolfram, 2002](#)] a été capable de monter un dérivé de la covariance de Lorentz à partir d'un modèle discret de réseaux causals trivalents, et les travaux de Bolognesi ont suivi cette voie, essayant de dériver des propriétés potentiellement émergentes à partir de réseaux causals [[Bolognesi, 2001](#)].

En science, on ne peut proposer outils et modèles que pour la prévision et la compréhension. Parfois, comme dans le cas de la mécanique quantique, la prévision est encore plus importante que la compréhension. Les mécaniques classique et relativiste, cependant, sont toutes deux déterministes, et donc théories de prévision (ou en d'autres mots, si on est incapable de prévoir en mécanique classique, c'est à cause d'un manque de précision dans les mesures). Donc, si le gros de la physique est déterministe, pourquoi serait-il difficile d'accepter qu'il n'existe que du pseudo-aléatoire dans notre univers ? La seule théorie physique courante qui suggère concrètement l'existence d'un aléatoire indéterministe est la mécanique quantique. Et nous comprenons si peu de choses à son sujet que l'étonnant serait qu'on ait véritablement à remercier la mécanique quantique pour donner au monde la seule manifestation de véritable aléatoire. De plus, une telle suggestion dépend d'une interprétation particulière de la mécanique quantique – une parmi plusieurs – ce dont on discutera plus tard.

Quand Leibniz a commencé à faire de l'arithmétique binaire, il pensait possible que le monde soit parvenu à exister à partir de rien (zéro) et un, car on peut rendre compte de tout dans ce langage, langage binaire (le plus simple langage possible basé sur le nombre de symboles impliqués). Sorti de ce langage extrêmement simple, né de sa beauté, le monde est venu au monde sous ses propres yeux. S'il était digital au plus bas niveau, on devrait finir par ne voir que des chaînes de bits. Elles auraient eu l'air très aléatoires durant les premières secondes de l'existence de l'univers, puis auraient soudain commencé à former des motifs équivalents aux amas de particules – et c'est d'ailleurs ce qui s'est passé. Les chaînes auraient formé l'équivalent des ga-

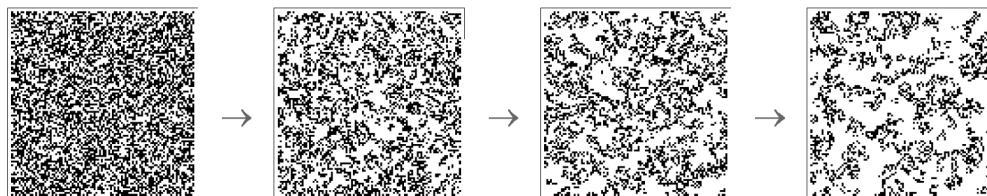


FIGURE 4.2 – Structures générées par un automate cellulaire 2D à 9 voisins, qui est la règle numéro 40 de Wolfram. Ici, on voit les diagrammes de l'état de l'espace toutes les 6 étapes, en partant d'un dispositif 100 x 100 bits (pseudo) aléatoires.

laxies, des chaînes de galaxies, des chaînes de systèmes solaires et des chaînes de planètes, des chaînes de toutes sortes de choses.

L'existence dans l'univers d'ordinateurs digitaux construits par les humains est une preuve évidente que l'univers est capable de réaliser des computations digitales. La question principale est si cette computation se produit naturellement dans l'univers, jusqu'où va-t-elle, et si elle est du même type que celle réalisée par les ordinateurs. Dit autrement, quelle différence entre un univers de chaînes et l'univers dans lequel nous vivons ? La version chaînes est une sursimplification, bien sûr, mais finalement les deux ne devraient pas être si différents que cela.

4.4 Les lois physiques comme des programmes informatiques

Les programmes informatiques sont comme les lois physiques ; ils produisent de l'ordre en laissant filtrer une partie de ce avec quoi ils ont été nourris. Commencez avec une chaîne qui a l'air bien aléatoire, faites tourner un programme choisi au hasard sur elle, et vous avez de grandes chances de voir votre chaîne à l'air aléatoire devenir une chaîne régulière, souvent non-triviale, et hautement organisée. Au contraire, s'il n'y avait pas de lois physiques et que vous en étiez réduit à jeter les particules en l'air, les chances qu'elles se regroupent de la façon dont elles le font seraient si minces, que rien ne se passerait dans notre univers. Les lois physiques sont comme des logiciels, elles font en sorte que quelque chose se passe.

π , par exemple, est plus simple à calculer en jetant en l'air des bits interprétés comme des instructions de programme informatique, peu importe

le langage (comme expliqué dans la section précédente). Mais c'est beaucoup plus dur, si on veut générer les digits de π , de jeter en l'air les digits eux-mêmes. C'est parce que les programmes de π seront toujours courts par rapport à leur expansion.

Tout comme les formules produisant les digits de π sont les versions compressées de π , les lois physiques vont distiller les phénomènes naturels à partir de séries d'observations. Ces lois sont valables car grâce à elles, on peut prévoir le résultat d'un phénomène naturel sans avoir à attendre qu'il se déroule en temps réel. Résoudre les équations décrivant le mouvement planétaire au lieu d'avoir à attendre deux ans pour connaître la position future d'une planète, on peut (presque¹) connaître leur évolution précise avec, en principe, une absolue certitude. Les lois sont toujours associées à des calculs, et ce n'est pas une coïncidence si tous ces calculs se trouvent être calculables, que les computations soient effectuées par des humains ou des ordinateurs. En pratique, les lois physiques, comme cela a été suggéré (par ex. Chaitin [Chaitin, 1975]), sont comme des logiciels compressés.

4.5 Pas de séparation entre programmes et données

Parmi les nombreuses contributions faites par Alan Turing basées sur son concept d'universalité computationnelle, il y avait l'unification entre données et programmes. Tandis qu'on peut considérer l'entrée fournie à une machine de Turing comme une donnée, et le tableau d'une règle de machine de Turing comme son programme, chacun comme des entités séparées, en fait elles sont interchangeables.

Si $\langle M \rangle$ est le codage binaire d'un programme décrivant une machine de Turing M , on peut construire une paire $(\langle M \rangle, x)$ pour la machine de Turing M dotée de l'entrée x et établir (comme Turing l'a fait lui-même) qu'il existe une classe générale de machines du même type (aussi des machines de Turing) qui émulent/imitent M pour l'entrée x , et sont capables d'accepter cette description $(\langle M \rangle, x)$ comme entrée et de l'émuler (de donner ce qu'aurait donné M avec x comme entrée), prenant ainsi $(\langle M \rangle, x)$ – qui est un programme plus une donnée – comme une simple donnée pour U .

1. Bien que ce soit un sujet entier à lui tout seul, on peut dire ici que si nos théories sont approximatives, c'est dû à la même brisure de symétrie apparaissant à toutes les échelles, et qui fait diverger nos prévisions sur le long terme.

La machine U s'appelle une machine de Turing universelle car elle peut imiter tout autre machine.

D'un autre côté, à partir de toute entrée fournie à M , on peut construire M' avec une entrée vide de façon à ce que M et M' produisent la même sortie. En d'autres mots, on peut toujours incorporer une donnée comme partie du tableau de règles d'une nouvelle machine. On considère par là une donnée comme un programme. Cela unifie programmes et données car elle peuvent toujours être utilisées de façon interchangeable.

C'est pourquoi aujourd'hui, on peut explorer l'univers computationnel soit en suivant une énumération des machines de Turing, soit en utilisant une machine (universelle de Turing) tournant sur une énumération de programmes comme données d'entrée. Les deux approches sont exactement semblables grâce à l'universalité de Turing.

L'identification de quelque chose en tant que donnée ou en tant que programme est simplement affaire de convention et d'habitude, ce n'est pas une distinction fondamentale. Il est important de l'avoir à l'esprit quand on considère une vision computationnelle de l'univers, car on peut assez raisonnablement se demander quelle partie représente les données, et quelle autre les programmes. La réponse habituelle est que les programmes sont les lois physiques, comme on l'a concrètement suggéré au début. Pourtant, on n'a pas vraiment besoin de faire une telle distinction, si, comme pour les machines de Turing, données et programmes sont de même type par nature, comme cela est suggéré par une vision computationnelle du monde.

4.6 La signification d'algorithmique

On peut objecter que nous avons été imprudents dans notre utilisation du terme algorithmique, en ne disant pas exactement ce qu'il signifiait. Dans notre contexte, algorithmique est l'adjectif utilisé pour signifier un ensemble de processus ou de règles capables d'être exécutés par un ordinateur, par opposition avec un processus véritablement (indéterministe) aléatoire (qui est incalculable). Les modèles classiques de computation² saisissent ce qu'est un algorithme, mais ce papier met expérimentalement en contact les significations en théorie et en pratique, essayant d'aligner les deux. D'un côté, on a le cadre théorique de la probabilité algorithmique. De l'autre, on a les données empiriques. Il n'y a pas moyen de les comparer entre eux à cause de

2. Quoi qu'en supposant la thèse de Church-Turing (qui dit que les fonctions calculables par algorithmes sont les fonctions récursives).

la non-calculabilité de la distribution universelle de Levin . On procède en construisant une distribution algorithmique expérimentale en faisant tourner des machines calculantes abstraites (donnant donc une distribution purement algorithmique), qu'on peut alors comparer à la distribution de données empiriques, pour trouver plusieurs types de corrélations avec différents degrés de signification. Donc pour nous, “algorithmie” signifie l’accumulation exponentielle de règles productrices de motifs, et la mise à l’écart des règles productrices d’aléatoire. En d’autres mots, l’accumulation de règles simples.

Dans notre contexte, quelque chose est algorithmique s’il suit une distribution algorithmique (par ex. la distribution expérimentale que nous avons calculée). On peut donc prendre cela pour une mesure d’“algorithmicité” : le degré d’approche d’un ensemble de données par rapport à une distribution algorithmique expérimentalement produite (supposée être une distribution de Levin). Plus on est proche d’une distribution algorithmique, plus on est algorithmique.

Donc quand on établit qu’un processus est algorithmique par nature, on veut dire par là qu’il est composé de règles simples et déterministes, des règles produisant des structures que prévoient théoriquement les probabilités algorithmiques. Ainsi, on peut voir la complexité algorithmique comme fournissant une mesure des ressources computationnelles nécessaires pour spécifier une structure.

4.7 Le modèle du calculateur quantique

Un ordinateur quantique diffère d’une machine de Turing au sens où ses bits sont des bits quantiques, et peuvent donc exister en état de superposition. De plus, on peut donner des instructions pour placer ces bits en superposition. Une machine de Turing universelle peut simuler toute autre machine de Turing en un temps polynomial, mais tandis que ce que réalise un ordinateur quantique est toujours Turing-calculable, un calcul quantique typique de T étapes sur n bits requiert $O(2^n)$ bits sur une machine de Turing classique, et $O(T2^{2n})$ opérations logiques ou étapes. Il faut donc des quantités exponentielles de temps et d’espace pour qu’une machine de Turing simule un ordinateur quantique ou une machine de Turing quantique.

L’ordinateur quantique standard comme l’a décrit Feynman, et plus tard Deutsch, diffère de l’ordinateur classique uniquement par rapport au concept de qubit (un bit qui est capable de se mettre en superposition quantique et donc de calculer en parallèle). Alors que le concept de qubit tire avantage

du net atout que lui donne le monde quantique – l'intrication – il ne fait pas nécessairement usage de toutes les propriétés de la mécanique quantique. La définition pratique d'un qubit ne fait pas qu'un ordinateur quantique calcule plus mais plutôt plus vite qu'un ordinateur digital classique selon la définition standard, mais en même temps selon l'interprétation de Copenhague, et contrairement aux ordinateurs classiques, la mécanique quantique est capable de produire un véritable aléatoire indéterministe.

Pour aider à faire avancer le débat, examinons l'inférence suivante, dont les prémisses sont (1) et (2) et la conclusion (3) :

1. l'univers est entièrement descriptible par la mécanique quantique
2. le calcul quantique standard intègre complètement toute la mécanique quantique
3. donc, l'univers est un calculateur quantique

Seth Lloyd prétend avoir prouvé la connexion entre 1 et 2, ce qui probablement met en relation isomorphe la théorie standard (ou un certain standard) de la mécanique quantique avec le modèle standard de calcul quantique.

La thèse de Lloyd ajoute une importante et remarquable affirmation (quoiqu'elle dépende de la conception du calculateur quantique) à la conception de l'univers en tant que Turing-calculateur, c'est-à-dire que non seulement l'univers est Turing-calculable, mais constitué de particules quantiques se comportant selon la mécanique quantique, il est un calculateur quantique.

Finalement, la définition rigide du qubit associée avec la diversité, la polyvalence des interprétations possibles que permet la mécanique quantique, rend difficile l'établissement des limites de l'affirmation que l'univers est un calculateur quantique. Si on suppose que c'est un calculateur quantique dans le sens de la définition du qubit, alors une description de l'univers en ces termes suppose que les particules quantiques codent seulement une quantité finie d'information, comme le fait le qubit, et que le qubit peut servir pour une pleine description du monde.

Cependant, le calcul quantique peut avoir une autre propriété pouvant le rendre plus puissant [Feynman et al., 2000] que les machines de Turing. C'est la production gratuite d'aléatoire indéterministe. Néanmoins, aucune interprétation de la mécanique quantique n'exclut la possibilité d'aléatoire déterministe, même au niveau quantique. Quelques collègues, cependant, ont obtenu certains résultats intéressants établissant que les théories à variables cachées devraient avoir besoin de beaucoup plus de ressources en mémoire

pour rendre compte de phénomènes quantiques connus. En d'autres mots, les théories à variables cachées sont plus coûteuses à soutenir, et la mémoire nécessaire pour simuler ce qui se passe dans le monde quantique croît tant et plus dans le cas de certaines machines déterministes. Mais encore une fois, cela ne permet pas d'exclure d'autres possibilités, ni même les théories à variables cachées, et un oracle quantique implique d'abord que la computation quantique est en fait indéterministe, et ensuite qu'on peut concrètement en faire quelque chose d'utilisable. Car si l'aléatoire indéterministe quantique est possible, il reste à expliquer comment cet aléatoire peut avoir un quelconque impact sur la mécanique classique (le premier, ou le plus important étant la divergence par rapport à la relativité). Par exemple, à une échelle quantique, ce qui gouverne les particules est réversible par rapport au temps, et toutes les lois physiques préservent une telle réversibilité par rapport au temps à un niveau quantique. Pourtant cette réversibilité temporelle semble n'avoir aucun impact sur notre réalité macroscopique, laissant supposer l'existence possible d'une déconnexion causale. C'est peut-être aussi le cas avec l'aléatoire quantique, s'il existe. Il se peut que l'aléatoire quantique n'ait aucun impact au-delà de l'échelle quantique. Pensons aussi à l'intrication ; jusqu'ici il a été assez difficile, et cela le restera peut-être, de tirer avantage des états quantiques superposés. Car lorsqu'une particule intriquée interagit avec le monde macroscopique, ses états s'effondrent vers un unique état déterminé et l'état quantique est perdu, déconnectant apparemment la mécanique classique de la mécanique quantique.

4.7.1 L'arc-en-ciel des bruits colorés en tant que pseudo-aléatoires.

Le bruit blanc est habituellement identifié comme indéterministe ou comme un exemple de véritable aléatoire (dans un certains sens intuitif), pourtant je pense que tout aléatoire équivaut à des motifs compliqués résultant de l'application de règles algorithmiques (bien que je sois tout-à-fait conscient que ce point de vue s'oppose à l'interprétation de Copenhague de la mécanique quantique – et quoiqu'il soit compatible avec d'autres interprétations).

La question des autres couleurs de bruit est très intéressante. Beaucoup de couleurs de bruit, comme le bruit rose (le bruit qui suit une distribution $1/f$ et différent au bruit blanc qui suit une distribution uniforme) suivent par exemple une distribution de fréquence en loi de puissance de façon assez organisée. Les distributions en loi de puissance sont souvent une indication

que la source n'est pas aléatoire par nature. Les distributions associées à des processus aléatoires sont typiquement uniformes ou gaussiennes. Alors que les théories de bruit rose (et des bruits d'autres couleurs) sont encore un domaine courant de recherche, leur distribution typique en forme de loi de puissance est compatible avec la distribution algorithmique empirique trouvée dans les distributions que nous générons à partir de sources algorithmiques (et compatibles avec la distribution théorique universelle des lois de puissance). Nos distributions obtenues en faisant tourner des processus informatiques génèrent à peu près le même type d'aléatoire avec à peu près la même fréquence. Je pense que ce type de bruit peut s'expliquer comme étant la queue de la distribution de probabilités algorithmiques, qui, bien qu'elle ait l'air aléatoire en grande partie, ressemble réellement à la partie haute plus organisée, que nous identifions comme la partie structurée correspondant aux signaux structurés.

Le bruit rose est présent un peu partout dans les séries de données. Il a tendance à apparaître dans des systèmes physiques naturels, depuis presque tous les appareillages électroniques jusqu'aux radiations électromagnétiques des corps astronomiques. Dans les systèmes biologiques, il est présent dans certaines statistiques sur les séquences ADN, une source que nous avons aussi analysée par rapport à ses distribution de motifs [[Bennett, 1997](#)] (triplets sur plusieurs longueurs de séquences) en montrant certaines corrélations avec les distributions algorithmiques que nous produisons à partir de machines abstraites et purement algorithmiques.

Supposer qu'on puisse compresser une donnée parce qu'elle fait des motifs, même si tout en elle n'est pas produit par des moyens purement algorithmiques, est une position raisonnablement défendable. Ce pourrait bien être le cas. Quelques chercheurs insistent sur le fait que des ensembles de données empiriques seraient réellement aléatoires algorithmiquement [[Sutner, 2011](#)]. Cependant, il y a d'un côté le fait que ce signal algorithmique est présent presque partout autour de nous, et qu'il peut concrètement être (partiellement) quantifié comme nous l'avons fait, avec des distributions de fréquences de motifs (Cette position, que des données sont algorithmiquement aléatoires, a été fortement contestée [[Zenil, 2010](#)]). Mais d'un autre côté, des algorithmes produisent ces motifs à l'air aléatoire, que nous pouvons ne pas reconnaître comme étant produits de cette façon, nous conduisant à penser qu'ils ne peuvent pas être produit avec des règles déterministes, bien que le cœur du problème vienne simplement de ce qu'on ne peut les déchiffrer. Si un univers déterministe était capable de produire de vrais bits aléatoires, ce qui serait évidemment un argument contre le déterminisme, en leur présence on serait prêt à accepter que c'est réellement ce qui se passe (comme la mé-

canique quantique semblerait l'impliquer). Si je considère cette possibilité, c'est seulement pour souligner le fait que la mécanique classique implique un univers déterministe tandis que la mécanique quantique est supposée être une source d'aléatoire libre, ce qui est l'une des plus profondes divergences entre ces deux théories dominantes. Je veux seulement insister sur le fait que si l'univers est déterministe, comme on peut penser qu'il est, alors il y a incompatibilité fondamentale.

4.7.2 Divergence des définitions analogiques

Un des problèmes qu'on rencontre quand on essaye de prendre une position par rapport d'un espace continu, c'est qu'il n'y a pas de moyen unique de déterminer le pouvoir de calcul dans un tel espace, et c'est l'un de mes arguments contre cette position, et en faveur d'une position algorithmique. En effet, on a un seul monde discret (sauf isomorphismes), mais un nombre infini de mondes continus ou "analogiques" (non pas banalement isomorphes).

Il y a certainement beaucoup de façons différentes de concevoir une définition de l'analogique. A partir de la Relativité Générale (GR), par exemple, on a de la matière qui doit couvrir un continuum (sinon la GR s'effondre en mécanique newtonienne), et comme telle, même s'il se trouve qu'on n'arrive plus à la diviser, la stratégie est de penser que la théorie de la relativité implique d'une certaine façon que la matière est infiniment divisible (juste une autre manière, peut-être incorrecte, de dire que cela couvre un continuum). Mais si ce qui est conçu comme un continuum se trouve être une conception abstraite d'un espace pris comme une entité indivisible, alors on se trouve avec une poignée de définitions possibles qui peuvent donner naissance à des conceptions complètement différentes et incompatibles d'un monde analogique (comme celle dans laquelle les choses peuvent se voir comme infiniment divisibles, comparée à celle où les choses ne peuvent se diviser mais sont physiquement reliées).

Comme Floridi [Floridi, 2004, Floridi, 2008] l'a suggéré, il est possible qu'on ne sache jamais si le monde est digital ou non. Le fait qu'on ne puisse pas se mettre d'accord sur ce que serait un monde analogique, alors que la vision digitale est à la base claire comme de l'eau de roche (dans un monde digital, la puissance de calcul est bien définie aussi) favorise l'hypothèse digitale. En montrant que quelque chose ressemble à quelque chose d'autre, on n'exclut pas les autres possibilités, et on réussit incroyablement mieux à modéliser le monde en utilisant les approches digitale et algorithmique. Pourquoi ne devrait-on pas croire que quelque chose est ce qu'il paraît être

plutôt que non ? Et quand je dis paraît, je pense à une liste plus longue que juste les propriétés sémantiques de ce mot. Je pense que quelque chose doit montrer statistiquement une certaine ressemblance avec quelque chose d'autre, et qu'il doit être ouvert à la quantification de la façon dont on l'a fait [Kolmogorov, 1965], (c'est-à-dire être capable d'être supporté par un calculateur analogique) tout en restant algorithmique.

Tout comme on peut penser à un objet continu de plusieurs façons, comme par exemple quelque chose qui ne peut se casser en parties individualisées, ou comme quelque chose qui soit intrinsèquement lié d'une certaine façon continue, le "discret" ne vient pas immédiatement de "l'algorithmique", bien qu'il puisse l'impliquer au moyen du digital. Sans tenir compte d'une mesure de complexité, les modèles continus vont nécessairement contenir plus de données. C'est pour cette raison qu'on peut invoquer le rasoir d'Occam quand on compare l'hypothèse digitale à l'hypothèse analogique.

4.7.3 Caractéristiques concevables pour les modèles de l'univers

Il est intéressant de voir comment des modèles physiques basés sur des théories mathématiques supposent des variables continues, alors que lorsqu'on résout les équations, disons celles de la relativité générale, le modèle devient algorithmique au sens digital le plus strict, que ce soit par la vertu des moyens mécanistes par le truchement desquels les équations sont résolues à la main, ou que ce soit littéralement quand elles sont résolues par un ordinateur digital. La vision algorithmique peut fournir un raccourci pour comprendre la nature digitale du monde sans avoir à supposer qu'il l'est au final.

Quand la relativité générale substitue la mécanique classique comme théorie des phénomènes à grande échelle, elle arrive équipée d'outils pour prouver ou réfuter sa validité. Pour mettre en place une analogie plausible avec ce développement, il faudrait d'abord démontrer les limites d'un modèle pour ensuite argumenter en faveur du pouvoir prédictif d'un modèle alternatif. Je trouve qu'il est regrettable que certains auteurs persistent à utiliser ce genre d'analogie par rapport et comme un argument en faveur de l'hypercalcul. Je fais référence à l'idée que, dans un certains sens, l'hypercalcul serait à la théorie classique du calcul ce que la théorie de la relativité est à la mécanique classique. Aucune chose de la sorte n'est arrivée à favoriser le calcul analogique par rapport au digital, et personne n'a non plus été capable de construire un hypercalculateur. Il n'y a ni progrès ni unification qui ressortent

de cette pratique consistant à proférer des analogies. La mécanique classique a rencontré le succès car elle fut capable de relier des phénomènes se produisant à la surface de la Terre avec le mouvement des planètes et d'objets dans le ciel (gravité). Puis la relativité générale a mûri jusqu'à englober ce que la mécanique classique était capable de modéliser, expliquer et prédire, à l'intérieur d'une théorie capable de décrire et de prédire des phénomènes à des échelles plus grandes et à des vitesses plus élevées. Mais rien de comparable ne s'est produit en calculabilité, où les modèles se sont tous retrouvés avec leurs pouvoirs computationnels équivalents, et ceux atteignant une puissance plus élevée ne l'ont fait que théoriquement. Le seul exemple concret, pratique et faisable de computation est celui qui a été réalisé avec des moyens digitaux, et même les calculateurs analogiques pourraient s'avérer ne contribuer à aucun pouvoir supplémentaire (mesurable). Même si la thèse de Church-Turing est fausse, ces analogies n'ont que peu de signification.

D'autres analogies malheureuses concernant l'hypercalcul ont été proposées dans le passé – avec les géométries non-euclidiennes. Mais celles-là aussi semblent inappropriées car les géométries non-euclidiennes sont hautement constructives, chose que l'hypercalcul ne peut être, en principe. En effet, ces analogies paraissent basées sur un morceau de raisonnement circulaire : ce qui semble être faisable par des moyens mécaniques, ou étape par étape, algorithmiquement, ou supporté par une machine de Turing, est constructif. Ce qu'il faudrait faire, c'est définir la constructivité en termes d'hypermachines de Turing, mais il faudrait d'abord prouver que de telles machines sont concrètement réalisables, tout comme l'est la version d'une machine de Turing construite comme calculateur digital. Je pense que la seule analogie valable avec les géométries non-euclidiennes, c'est que tout comme on dérive l'une ou l'autre géométrie non-euclidienne grâce à un processus de négation (le 5ème postulat), le démenti de la thèse de Church-Turing pourrait ouvrir la porte à l'hypercalcul. Mais les partisans de ces analogies paraissent s'en servir plus comme un moyen d'attirer l'attention sur l'injustice dont ils se considèrent victimes, que pour de strictes raisons techniques. Le sentiment général dans le domaine est que le reste de la communauté scientifique de l'informatique théorique est incapable de reconnaître la valeur de l'hypercalcul. Malheureusement pour les hypercalculistes, cela n'arrivera pas tant qu'ils ne se présenteront pas avec un hypercalculateur, alors la communauté fera attention à eux. Pour le moment, ils ont un formidable défi à relever. Ils ne doivent pas seulement se présenter avec un appareillage concret, mais aussi avoir la preuve qu'un tel montage est capable d'hypercalcul. Et la reconnaissance de l'hypercalcul va finir par être équivalente à la construction de l'hypercalculateur lui-même, car seul un hypercalculateur peut être capable

d'en reconnaître un autre.

4.7.4 Hypothèse de l'aléatoire contre hypothèse de l'ordonné

Le but des théories scientifiques est d'expliquer les structures présentes dans les données. Comme Calude et Salomaa l'ont observé [Calude and Salomaa, 1994], la science est fondée sur ce principe – la supposition que l'univers physique est ordonné et contient des régularités pouvant être saisies par des modèles et des lois. Chaitin a suggéré que les lois physiques (et donc l'entreprise de la science dans son entier) peuvent se voir comme un logiciel, avec les observation empiriques jouant le rôle d'entrées, tandis que les lois, fonctionnant comme le font les programmes informatiques, vont générer des sorties sous la forme de prévisions. Cette approche a rencontré un vif succès dans l'application des maths et des algorithmes pour réaliser des prévisions exprimées en termes de lois et de principes physiques. Une théorie scientifique est, d'une certaine manière, une distribution de probabilités expliquant une distribution de données empiriques. Car il se trouve que si on dispose d'une théorie et de certaines données empiriques, alors on peut utiliser notre connaissance de la théorie pour coder les données sous une forme compressée.

Calude et Salomaa se servent de la théorie de l'information algorithmique pour mettre à l'épreuve cette hypothèse en utilisant l' Ω de Chaitin comme un modèle de l'univers. Ce qui est très intéressant, c'est que le modèle de Calude et Salomaa est un modèle discret, pourtant il ne met pas en opposition un univers aléatoire face à un univers algorithmique. Cependant, un problème courant est le problème de la sur-adaptation, faisant qu'un modèle faux (mais infalsifiable) peut s'adapter parfaitement avec un phénomène observé. Une comparaison statistique ne peut pas concrètement être utilisée pour catégoriquement prouver ou réfuter une différence ou une similitude, on ne peut que favoriser une hypothèse par rapport à une autre.

Comme cela a été prouvé par Levin, la mesure de la probabilité algorithmique (la distribution universelle) va surclasser toutes les autres mesures, à moins que d'autres informations soient disponibles pour aider à prévoir le résultat, auquel cas une variable supplémentaire doit être ajoutée au modèle pour tenir compte de cette information. On a prouvé que l'induction algorithmique de Solomonoff, même si elle n'est pas calculable bien qu'on puisse l'approcher, est un prévisionneur optimal [Solomonoff, 1964], en un certain sens.

On sait que la probabilité algorithmique repose sur deux principes essentiels : le principe des explications multiples, qui déclare qu'on doit retenir toutes les hypothèses consistantes par rapport aux données, et un second principe connu sous le nom de rasoir d'Occam (le rasoir d'Occam est précisément ce que la distribution de Levin formalise), qui déclare que lorsqu'on mobilise des causes, les entités mobilisées ne doivent pas être multipliées au-delà du nécessaire, ou dit autrement, que parmi toutes les hypothèses consistantes par rapport à un ensemble d'observations, les plus simples doivent être favorisées. Comme pour le choix d'une distribution a priori bâtie sur une hypothèse, cela revient à assigner une plus haute probabilité aux hypothèses les plus simples, et une plus basse probabilité aux hypothèses les plus complexes. C'est là que le concept de complexité algorithmique entre en scène.

Un univers analogique pourrait être incompréhensible, inconnaissable, indétectable et indescriptible. Si le monde est digital ou algorithmique, il n'est pas beau parce qu'on peut le comprendre entièrement, mais plutôt parce que, contrairement au monde analogique, il est compréhensible en principe (même si en fait il ne l'est pas et ne le sera jamais jusqu'au bout). Très grand ne signifie pas infini, tout comme fini ne veut pas dire petit, trivial ou facile. Le monde peut encore être un endroit très intéressant même s'il est discret et déterministe. Wolfram [[Wolfram, 2002](#)] a empiriquement exploré le monde de Turing, ce qu'il appelle l'univers computationnel, et il l'a trouvé extrêmement riche, au moins autant que les processus du monde réel (analogique ou non). Si le monde est analogique, jusqu'à quel point est-il analogique ou suffisamment puissant pour l'être ? Il n'y a pas d'accord sur un modèle général de calcul analogique, et il se pourrait bien qu'il ne puisse pas y en avoir, car au-dessus d'un monde analogique donné, il y en aura toujours un autre, plus compliqué. Le monde digital, cependant, ne peut être qu'unitaire en quelque sorte, car tous les modèles digitaux de l'univers sont de puissance computationnelle égale, tous autant sophistiqués les uns que les autres, et constitués ou décrits par les mêmes unités de base : des bits. Si le monde, dans toute sa richesse des possibles, était analogique, il serait composé de choses disparates nouant des relations compliquées avec d'autres choses. Certains mondes auraient la puissance du monde digital, tandis que d'autres pourraient avoir n'importe quel autre puissance concevable. Si c'était le cas, il faudrait se poser des questions au sujet de l'efficacité des computations digitales expliquant le monde (calculant avec lui et faisant en sorte qu'il calcule). Un monde digital permettrait d'avoir une identité théorique entre une simulation par ordinateur et son instantiation physique, au sens où tous les états de l'instanciation peuvent être modélisés étape par étape pour une simulation digitale. La question reste ouverte de savoir si notre capacité à simuler le monde (ou

à échouer à le faire avec précision) est une indication en faveur de l'une ou l'autre possibilité. Un des nombreux avantages d'une ontologie digitale est qu'un modèle digital serait capable de simuler pleinement le monde. Ce qui ne veut pas dire qu'il serait capable, par là, de prédire ou de comprendre pleinement tous les processus, mais cela signifie qu'au moins en principe, on pourrait rester aligné sur la réalité. Si le monde était analogique, il n'y aurait pas de façon évidente de savoir si une simulation analogique serait capable de rester alignée avec le monde analogique, d'abord parce que les deux pourraient ne pas converger du point de vue de la puissance computationnelle. Un autre avantage est que les calculateurs digitaux permettent, par exemple, l'existence de compilateurs, qui sont des traducteurs entre différents calculateurs. L'universalité de Turing garantit qu'on peut toujours traduire tout programme donné en un programme Turing-complet.

On peut résumer l'importance du modèle digital de computation avec une citation de Gödel (dans une conférence donnée juste après une donnée par Tarski).

Dans sa conférence, Tarski a mis l'accent (je pense avec raison) sur la grande importance du concept de récursivité générale (ou calculabilité de Turing). Il me semble que cette importance est largement due au fait qu'avec ce concept, on a pour la première fois réussi à donner une définition absolue d'une intéressante idée épistémologique, c'est-à-dire quelque chose qui ne dépend pas du formalisme choisi. (Gödel 1946)

Ce que je pense que Gödel voulait dire, c'est que le modèle de Turing semble assez général pour être pris comme la base de la base, et qu'on peut donc supposer qu'il n'existe rien de spécial dans la description de sa formalisation.

Les calculateurs digitaux sont souvent mis en contraste avec les calculateurs analogiques, qui sont des machines manipulant des variables continues, par ex. le temps et l'espace, contrairement à la computation digitale dans laquelle temps et espace sont discrets. Néanmoins, si le monde est un monde analogique, il y aurait de grandes chances pour qu'il ressemble à un Ω de Chaitin, un nombre aléatoire par définition dépendant de l'imprévisibilité des calculateurs digitaux universels, appelé aussi la probabilité d'arrêt. Le monde peut être soit un petit morceau apparemment ordonné au milieu d'un univers potentiellement entièrement aléatoire, comme Calude et Meyerstein le suggèrent [Calude and Meyerstein, 1999], auquel cas nous serions très chanceux de vivre dans le minuscule fragment apparemment ordonné, soit ce serait un monde structuré, comme il semble l'être.

La question me paraît être de savoir si on peut associer l'aléatoire avec le concept d'un monde analogique. Bien que la relation ne soit pas triviale, ni que l'aléatoire soit la définition de l'analogique, je pense qu'on associe souvent l'analogique à des degrés plus ou moins grands d'incertitude. Que ce soit sous la forme d'indéterminisme vrai ou sous la forme de défauts fondamentaux faisant obstacle à une infinie précision des mesures, je serais tenté de lier ces deux formes avec des propriétés associées à la fois à l'analogique et à l'aléatoire. Par exemple, dans les systèmes dynamiques, le chaos aléatoire est habituellement défini comme l'infini des trajectoires possibles le long desquelles un système peut diverger au cours du temps et de l'espace, en partant de configurations initiales très proches. Et si le monde était analogique, on serait aussi fondamentalement incapable de faire des mesures tout-à-fait précises, comme s'il y avait l'implication d'un certain aléatoire (au niveau des l'incertitude des mesures).

Si le monde est un monde analogique, alors on doit vivre dans le meilleur monde analogique possible. C'est ce que je prétends, en l'occurrence que les chances de trouver des motifs et des structures dans un monde analogique seraient très basses à moins qu'on suppose que notre monde soit le meilleur parmi tous les mondes possibles. Cependant, sous la vision digitale, motifs et structures sont à la base une conséquence inévitable. Si notre monde est digital, alors notre monde est par essence comme n'importe quel autre, tout particulièrement au sens algorithmique.

Comme l'ont prouvé Solomonoff et Levin , toute autre supposition, si elle concerne le produit d'une machine de Turing, va simplement passer outre la somme des probabilités algorithmiques. Donc plutôt que d'être plus précis, tout autre modèle va différer des probabilités algorithmiques au sens où il va nécessairement finir par négliger une partie des données. En d'autres mots, il n'y a pas meilleur modèle pour prendre en compte toutes les données que le modèle des probabilités algorithmiques, si elles ne sont pas aléatoires. Comme Solomonoff l'a affirmé, on ne peut pas faire mieux. L'inférence algorithmique est un problème d'optimisation d'un temps limité, et les probabilités algorithmiques en tiennent simplement compte.

4.8 Une interprétation algorithmique de la mécanique quantique

Prendre quelque chose comme modèle d'un objet ou d'un phénomène ne signifie pas que cette chose est réellement l'objet ou le phénomène. Même si

à des échelles plus grandes que les distances atomiques les modèles peuvent être hautement précis, ils ne décrivent pas nécessairement l'univers à toutes les échelles ou dans tous les cas, ce qui doit nous rappeler que les modèles ne sont pas toujours de pleines descriptions de la réalité.

Bohr a dit un jour [[Bohr, 1934](#)] :

... [dans] la description de la nature, notre but n'est pas de découvrir l'essence réelle des phénomènes, mais seulement de dépister, aussi loin que possible, les relations entre les multiples aspects de notre expérience. A strictement parler, le formalisme mathématique de la mécanique quantique et de l'électrodynamisme offre simplement des règles de calcul pour déduire ce à quoi on doit s'attendre quant aux observations obtenues sous des conditions expérimentales bien définies spécifiées par les concepts de la physique classique.

La mise au rebut d'explications qui étaient auparavant des lois est une pratique courante, sinon le but même de la science, et on tend à les remplacer par des modèles qui expliquent de plus en plus de phénomènes et possèdent un plus grand pouvoir prédictif. Bien sûr, dire que le monde est quelque chose ne va pas le transformer en cette chose, mais quand quelque chose a la même odeur, le même aspect et le même comportement qu'une chose spécifique, on peut être convaincu que c'est effectivement cette chose-là. Dans ce cas et selon le plus extrême scénario, il semble clair que l'information soit quelque chose d'aussi fondamental que d'autres variables du monde physique, telles que matière et énergie (même si aller aussi loin correspond à épouser un point de vue sur le monde, et pas nécessairement établir une vérité ontologique). Que ce soit vrai ou non, cela reste à voir, ou peut-être ne le saura-t-on jamais. A noter quand même que de nombreux physiciens en sont arrivés aux mêmes conclusions à cause du concept de symétrie (pouvant se voir comme une information ou un objet mathématique abstrait), qui joue un rôle de base, même dans la prédiction de l'existence de nouvelles particules.

Et jusqu'ici, cette approche a eu un certain succès dans de nombreux domaines de la physique. Il est remarquable de voir comment une simple description informationnelle de la mécanique quantique suffit à décrire complètement des phénomènes quantiques. Ces descriptions spécifient comment les particules interagissent et comment elles sont reliées les unes aux autres par des symétries.

4.9 La place de l'information dans le réel

Si toute la matière est constituée des mêmes particules de base, qu'est-ce qui fait qu'un objet soit différent d'un autre à part le fait qu'il occupe un espace différent ? Ce qui fait qu'une tasse soit une tasse et non un être humain est simplement la façon dont les particules sont configurées. On peut démanteler une tasse (ou plusieurs) pour les réassembler en être humain. Ce qui fait qu'une tasse soit une tasse et un humain soit un humain, c'est l'information. L'affirmation disant que seule l'information fait qu'une tasse soit une tasse et non un humain, est basée sur le fait que tasses et humains sont constitués exactement des mêmes particules élémentaires, seule la façon dont elles sont arrangées fait qu'on ait les unes ou les autres.

En fait, c'est déjà un peu comme cela avec les cellules de notre corps qui sont remplacées plusieurs fois au cours de la vie, alors qu'on reste plus ou moins les mêmes. C'est parce que ce qui fait que nous soyons ce que nous sommes n'est pas la matière avec laquelle nos organes et cellules sont constitués, c'est plutôt la façon dont ces cellules, et tout le reste de notre corps, sont configurées de manière précise, sur la base d'informations transmises au cours du temps. L'information perdue lors de la mort d'une cellule est récupérée à partir d'une autre et remise en place (presque) comme auparavant. La différence entre l'original et la copie est ce qu'on identifie comme le vieillissement, et certains biologistes pensent que c'est le résultat d'erreurs produites par l'environnement (par ex. les oxydants), tandis que d'autres biologistes ont suggéré que le corps est programmé pour vieillir, de façon à ce qu'une génération meure pour laisser la place à une nouvelle, permettant ainsi à la sélection naturelle d'opérer.

Dans un univers informationnel, nos corps et le monde se calculeraient eux-mêmes, rendant les choses capables de mémoriser leurs identités. Produire des bits aléatoires dans un univers discret, où tout événement est à la fois cause et conséquence d'autres événements, serait concrètement très coûteux, car il faudrait concevoir un moyen de briser le réseau causal, à supposer qu'il soit possible de commencer par produire un bit aléatoire, tout en gardant le reste du réseau causal intact (sinon il n'y aurait rien d'autre que de l'aléatoire, ce qui n'est pas le cas). On pourrait alors produire des bits aléatoires à volonté à partir de processus physiques.

4.10 La nature algorithmique du monde

C'est peut-être simplement humain de penser que tout est piloté par des algorithmes (simples), parce que c'est notre façon de comprendre les choses quand on est confronté à des phénomènes naturels. Cependant, c'est aussi parfaitement raisonnable de penser que le monde est basé sur des règles simples, comme il se trouve l'être, qu'il est largement compréhensible et qu'il peut se comprendre en utilisant des modèles simples (des théories scientifiques actuelles obéissant à des formules simples prévoient toutes sortes de phénomènes naturels). Si ces règles simples arrivent à produire la complexité visible autour de nous, je pense qu'on peut supposer sans crainte qu'elles peuvent bien être responsables de l'organisation de la complexité dans le monde. Bien sûr, il se pourrait que la nature nous trompe, nous faisant croire que les règles sont simples alors qu'en fait elles seraient très compliquées, malgré l'apparence du contraire. La caractéristique la plus commune d'une théorie de base est sa capacité d'unification, même sans qu'elle fournisse nécessairement toutes les réponses à toutes les questions. Toute théorie aspirant à jouer le rôle de modèle pour l'univers doit couvrir un grand nombre de phénomènes empiriques auparavant expliqués ou non, tout en minimisant le besoin de suppositions nouvelles, par exemple en étant concrètement dérivée de ce qu'on avait seulement supposé une fois ou considéré comme accessoire.

Il y a aussi un problème de niveau d'explication. Comme la question que se pose Tommaso Bolognesi [[Bolognesi, 2001](#)], serait-il avisé de dire que la Théorie Quantique du Champ est fausse parce qu'elle ne prévoit pas l'existence d'organismes unicellulaires? Pareillement, serait-il sage de dire que la découverte de l'ADN est erronée car celui-ci ne peut (encore) expliquer le déploiement des protéines? Il le pourra peut-être un jour, ou peut-être pas, mais qu'il puisse ou non le faire n'a rien à voir avec une quelconque fausseté, c'est plutôt une question de niveau de sophistication des résultats, même si la configuration initiale donnait la codification de l'ADN, avec les multiples couches de processus superposées (partant des groupements fonctionnels qu'on identifie comme les gènes).

Inspiré par les classes de comportement de Wolfram, Bolognesi récapitule les types de phénomènes qu'une prise en compte digitale-computationnelle pourrait aisément expliquer en termes de comportement qualitatif dynamique observable dans notre monde physique :

- comportements apparemment aléatoires
- périodicité, et coexistence de structures régulières-périodiques et apparemment aléatoires

- auto-replication
- structures périodiques localisées interagissant les unes avec les autres, comme les diagrammes de dispersion de particules

Tout cela peut s'observer par exemple chez les automates cellulaires comme Wolfram l'a identifié [Wolfram, 2002], ainsi que dans les ensembles causals algorithmiques [Bolognesi, 2001] (une approche de la gravité quantique qui suppose que l'espace-temps est fondamentalement discret et que les événements sont partiellement ordonnés), en tant que propriété émergente (inattendue) plutôt que par le codage d'une quelconque évolution particulière des systèmes.

Le fait que les modèles de computation se soient avérés être profondément équivalents, pas dans leurs détails mais dans leur puissance de calcul, signifie qu'on peut vraiment écrire n'importe quel programme dans n'importe quel modèle. Les implémentations d'une machine de Turing dans des langages de programmation structurés ou de bas niveau peuvent nécessiter plusieurs lignes de code. Si une telle convergence computationnelle n'existait pas ou si cette convergence n'avait pas de correspondance physique, on aurait déjà dû rencontrer des régions calculant à des niveaux si étranges pour nous qu'on ne pourrait même pas les reconnaître comme modèles computationnels (et c'est peut-être le cas). Mais en l'occurrence, on n'a pas trouvé un tel éventail d'intelligences différentes (sur notre planète ou au-delà), seulement des variations de degré d'intelligence. Les animaux manifestent un éventail d'intelligences différentes mais elles semblent couvrir un intervalle, sans dépasser une limite supérieure. Et la réponse évidente est peut-être qu'il n'existe pas beaucoup de possibilités différentes, et que seul un petit nombre ou même un seul type d'intelligence est possible. S'il n'y en a qu'un seul, ce peut être n'importe lequel, mais il se trouve qu'on connaît très bien un type d'intelligence, un type que nous avons artificiellement développé nous-mêmes, qui est l'intelligence computationnelle (digitale). Et il s'avère que sa puissance est une puissance que nous ne considérons pas complètement étrangère à notre propre intelligence mais plutôt comme un sous-type de celle-ci (et peut-être avons-nous raison). Ce type d'intelligence peut paraître plus primitif que le nôtre mais son potentiel pour atteindre presque n'importe quelle capacité humaine est clair aujourd'hui, réalisant de meilleures performances parce qu'il est plus rapide, et pas nécessairement parce qu'il possède une puissance spéciale.

Selon Wolfram, l'intelligence est une question d'identification plutôt que de sophistication. Cela découle de son Principe d'Equivalence Computationnelle (PCE). Son PCE dit que la plupart des computations non-triviales s'avèrent posséder une sophistication équivalente. Cela peut expliquer par exemple pourquoi nous sommes incapables de maîtriser la prévision météo,

car finalement la météo est aussi sophistiquée que nous le sommes (que nos esprits le sont) et nous n'avons aucun moyen de court-circuiter les computations météorologiques, malgré l'utilisation de supercalculateurs pour faire des prévisions deux jours à l'avance (au plus) – qui sont souvent fausses le jour suivant. Donc, selon Wolfram, le temps est un système intelligent avec lequel on ne peut pas interagir car nous sommes une intelligence d'un type différent.

Selon le PCE de Wolfram, la mémoire est ce qui rend un système intelligent, car à la base, tout système qui ne se comporte pas de façon éminemment triviale va probablement être aussi sophistiqué que le plus sophistiqué des systèmes. Ainsi, quelle est la différence entre un système potentiellement intelligent et un autre qui montre des signes concrets d'intelligence ? C'est en quelque sorte une question d'échelle sur plusieurs directions. Quand Wolfram se demande si les nuages sont intelligents selon son principe, sa réponse courte est "oui". Ce qui expliquerait pourquoi à chaque fois que les humains désirent faire une prévision sur une pluie future, cela s'avère incroyablement difficile à réaliser sur plus de deux jours, et les prévisions météo sont souvent fausses, même d'un jour à l'autre. Comment se fait-il que cette prévision du temps soit si difficile malgré notre longue expérience de genre de prévisions ? Eh bien, les nuages se calculent eux-mêmes, et en tant qu'élément météorologiques, ils sont assez complexes, aussi complexes que des systèmes comme les cerveaux humains selon le PCE.

Le PCE a d'intéressantes implications pour l'intelligence artificielle car cela signifie que nous sommes entourés d'intelligence. Nous nous considérons comme essayant de créer péniblement des systèmes intelligents en tant que tels, alors qu'en fait ce qu'on essaye de fabriquer, c'est une intelligence que nous puissions nous-mêmes reconnaître comme telle, par ex. une intelligence de notre propre type. Mais la conclusion intéressante est que nous n'avons pas vraiment besoin de faire tant d'efforts pour concevoir un système intelligent ; on pourrait simplement se servir d'un parmi les nombreux qui existent déjà dans l'univers computationnel, et peut-être le faire se comporter comme on veut (si on veut qu'il se comporte comme un être humain), pour cela il faudrait seulement que le système interagisse avec nous d'une façon ou d'une autre (et donc acquière une expérience sensorielle du même type).

Le dicton de John Wheeler [Wheeler, 1990] "ça vient d'un bit" (it from bit) représente la conception précoce d'un décalage digital vers une compréhension de la physique en terme d'information. Shannon [Shannon, 1948], avec sa théorie de l'information-communication basée sur des bits ; Turing [Turing, 1950], avec sa formulation d'un calcul sous forme de donnée et d'une donnée sous forme de calcul, et son concept de calcul universel ;

Landauer [[Landauer, 1961](#)] et Bennett [[Bennett, 1988](#)] , avec les définitions précises qu'ils ont donné aux connexions profondes entre information, calcul et principes thermodynamiques ; Levin [[Zvonkin and Levin., 1970](#)] et Chaitin [[Chaitin, 1975](#)], avec leur concept d'information algorithmique et probabilité computationnelle, nous ont fourni, avec d'autres, de précieux outils pour comprendre et réaliser de profondes relations entre réalité et calcul.

Je rappelle que Feynman [[Feynman et al., 2000](#)] disait en 1964 (citation tirée de l'une des six Messenger Lectures qu'il donna à l'Université Cornell) :

Cela m'a toujours embêté que, selon les lois telles que nous les comprenons aujourd'hui, une machine calculante ait besoin d'un nombre infini d'opérations logiques pour expliquer ce qui se passe dans une minuscule région de l'espace, aussi petite soit-elle, et durant une infime parcelle de temps, aussi courte soit-elle ... Ainsi, j'ai souvent fait l'hypothèse que finalement, la physique n'aura pas besoin de déclaration mathématique, que finalement on lèvera le voile sur toute la machinerie, et que les lois vont s'avérer être simples, comme le sont celles d'un plateau de jeu d'échecs avec toutes ses complexités apparentes.

Comme Wolfram l'a anticipé [[Wolfram, 2002](#)], et contrairement à certains physiciens qui semblent penser que la théorie expliquant l'univers sera finalement très compliquée et mathématique (voir, par exemple, le point de vue exprimé par Stephen Weinberg dans une récente interview avec Amir Aczel [[Weinberg, 2010](#)]), nous pensons que les corrélations trouvées peuvent s'expliquer comme suit : les processus physiques généraux sont dominés par des règles simples algorithmiques, les mêmes règles que les calculateurs digitaux sont capables d'exécuter.

La théorie de l'information de Shannon fait correspondre bits et matière (en l'occurrence avec de l'électricité passant dans un conducteur). Mais il semble aussi qu'un bit puisse être à la fois matériel et immatériel : pour Wheeler, il n'est pas constitué de matière, alors qu'il l'est pour Shannon. Un bit n'est pas une particule de réalité isolée et unitaire, comme le dicton "ça vient d'un bit" pourrait le laisser croire. Un bit peut être constitué de matière ou non, étant plutôt le résultat d'un processus, un processus d'interaction entre entités matérielles. Cependant, dans les deux interprétations, un bit ne peut pas exister sous une forme isolée (de la matière), car soit il n'existerait pas, soit il ne pourrait pas avoir de signification, car s'il y a absence de matière, aucune interaction n'est possible. Mais si l'information est liée à la matière, elle ne se cantonne pas à une relation en tête à tête avec elle. Un

spin, par exemple, est une position de la matière, alors qu'un trou dans une carte perforée est lié à de la matière sur une échelle plus grande, et au plus petit niveau atomique, la relation entre matière et information peut ne pas être représentable.

4.11 Comment la mécanique quantique implique le déterminisme

Il y a une bonne raison pour laquelle les chercheurs ont essayé par tous les moyens d'imposer une interprétation de la mécanique quantique où la seule issue possible est l'aléatoire indéterministe, en ne supposant rien d'autre que l'aléatoire indéterministe.

Quand la mécanique quantique fut lancée dans le cadre des travaux de Planck, Einstein et Bohr, il était clair que la physique classique ne pouvait pas expliquer des phénomènes à petite échelle comme l'interaction entre particules. La théorie de la mécanique quantique le pouvait, mais cela avait un coût très élevé car il fallait être prêt à faire toutes sortes de concessions par rapport à certains phénomènes incompatibles avec la physique classique. En l'occurrence l'intrication de particules, des particules qui sont dans tous les états possibles et à plusieurs endroits à la fois ; il y avait aussi le comportement dual de ces objets, se comportant parfois comme des particules, parfois comme des ondes ; et bien sûr les limitations concernant les mesures étant donné que ces mesures perturbent les systèmes en les faisant s'effondrer vers un mode classique.

Pour Bohr, ces concessions constituaient la réalité au niveau du monde quantique, alors qu'Einstein pensait que le caractère étrange de la mécanique quantique signifiait que la théorie était incomplète, et qu'elle finirait par perdre ce caractère incompatible avec la réalité familière que nous vivons. Il était convaincu qu'une théorie des variables cachées, encore à découvrir, serait capable d'expliquer certains, sinon tous ces étranges phénomènes.

L'interprétation de Bohr de la mécanique quantique est connue aujourd'hui sous le nom d'interprétation de Copenhague, c'est l'interprétation dominante actuellement. Elle établit que le monde quantique est à la fois ondulatoire et corpusculaire, et que c'est l'acte de mesure qui cause la manifestation d'un état ou de l'autre, que l'état d'intrication du monde quantique est réel, chose qu'Einstein n'a jamais acceptée (comme on peut s'en rendre compte avec sa fameuse déclaration disant que "Dieu ne joue pas aux dés"). [Plus exactement, "le Vieux ne joue pas aux dés" - NdT]. En fait l'intrication est

juste un mot pour indiquer la corrélation quantique. Selon l'interprétation de Copenhague, une paire de particules intriquées existe dans une superposition d'états quantiques jusqu'au moment de la mesure, à ce moment-là, la particule s'effondre dans un état spécifique sans relation avec aucun de ses états précédents, faisant ainsi d'elle une source intrinsèque d'aléatoire indéterministe.

Depuis lors, les physiciens ont mis au point des expériences pour éprouver la réalité quantique. La première fut montée par John Bell du CERN, montrant que si on suppose la corrélation des particules au sens où la mesure des propriétés de l'une va nous donner aussi les propriétés de l'autre, cela ne devrait pas marcher et on devrait se retrouver face à une inégalité décrivant la corrélation maximum possible dans un monde classique. Expériences après expériences, cette inégalité a été violée par la mécanique quantique, indiquant ce qu'on identifie habituellement comme deux particules intriquées. Cela signifie que deux suppositions faites par l'inégalité doivent être fausses, soit le réalisme, soit le localisme soit les deux. Le réalisme sous-entend l'idée que les propriétés des objets sont les mêmes en tous temps et que cela ne dépend pas de la personne observant l'objet ou mesurant ses propriétés. Le localisme suppose que ces propriétés ne peuvent pas être influencées par quelque chose d'éloigné et physiquement déconnecté. Cependant l'expérience n'exclut pas la possibilité d'une théorie à variables cachées, qui est une théorie classique où le réel et le local sont vrais.

Pour que l'expérience de Bell valide la réalité quantique, il faudrait concevoir et vérifier toutes les impasses et les failles possibles. Se peut-il qu'il y ait eu certaines erreurs introduites par les expériences de Bell faisant en sorte que les particules les plus corrélées soient celles qui avaient le plus de chances d'être mesurées? Cela expliquerait la violation. Alain Aspect a conçu une expérience pour tester et exclure ce cas. Cela implique la mise en œuvre de plusieurs détecteurs mesurant un grand nombre de paires de photons de façon à obtenir des résultats statistiquement significatifs. Après plusieurs expériences, la communauté en est arrivée à se convaincre que les expériences d'Aspect avaient exclu la possibilité que les expériences de Bell aient pu être statistiquement biaisées en ne mesurant que les particules corrélées. Se peut-il que ces expériences soient encore capables de nous tromper, nous faisant croire que la réalité quantique se comporte d'une certaine façon alors que ce n'est pas le cas?

Aspect a réussi à exclure le problème traitant le biais local dans la mesure de particules corrélées, mais qu'en est-il si quelque chose avait influencé l'expérience en communiquant les propriétés d'une particule à l'autre? Une façon d'exclure cette possibilité est de mettre de la distance entre les parti-

cules corrélées, assez pour garantir, grâce à la vitesse de la lumière, que si les mesures effectuées aux deux extrémités étaient suffisamment rapprochées dans le temps, rien ne pourrait voyager pour communiquer quoi que ce soit d'une particule à l'autre. Rappelons-nous que la supposition ici est que même si les deux particules sont corrélées car elles viennent d'une source unique, elles ne peuvent pas être corrélées au-delà de ce que l'inégalité de Bell établit, et que si la corrélation est plus grande que cela, cela veut dire qu'on a affaire à une inquiétante intrication entre les particules quantiques menant l'une d'elles à [Greenberger et al., 1990] ou à changer ses propriétés une fois que l'autre a été mesurée.

À l'évidence, si deux de ces particules sont assez proches pour communiquer entre elles, la violation de l'inégalité de Bell s'explique aisément. On suppose que l'information ne peut être disséminée sans support physique, un support matériel qui ne peut voyager plus vite que la vitesse de la lumière. Ainsi en plaçant les deux particules suffisamment éloignées l'une de l'autre, on peut exclure cette possibilité (encore une fois s'il n'existe pas de couche sous-jacente d'information, une réalité des variables cachées stockant ou communiquant les valeurs mesurées). La disponibilité de sources hautement efficaces de particules intriquées (avec l'émergence de la technologie laser) a permis aux scientifiques de réaliser les expériences d'Aspect avec un écart assez grand pour fermer la faille au sujet de l'information communiquée d'une particule à l'autre.

Maintenant cela ne peut fonctionner que si le réglage des détecteurs est modifié à chaque fois, étant donné que les expériences précédentes peuvent les avoir influencés, ou que l'un d'entre eux peut avoir trouvé le réglage de l'autre. Car pour garantir la validité d'une corrélation résultante des mesures, il faut qu'il n'y ait aucune mesure corrélée au niveau de la sortie. Weihs et Zeilinger ont réussi à séparer de 400m deux particules, en leur donnant 1,3 microsecondes pour modifier aléatoirement le réglage du détecteur, et un maximum de 5 nanosecondes entre les mesures garantit qu'aucune information voyageant à la vitesse de la lumière entre les deux extrémités ne puisse être transmise. Même quand certaines failles se referment dans les tests sur la mécanique quantique, d'autres peuvent s'ouvrir (par ex. la localité de l'effondrement) et aucun test n'a mis à l'épreuve toutes les failles en même temps. La théorie locale-réaliste des variables cachées, préférée d'Einstein, peut ne pas être une description viable du monde, mais il existe d'autres failles à refermer avant de prétendre qu'une interprétation quantique donnée est réelle à tous points de vue.

4.11.1 La question de la liberté de choix

Il y a une faille qu’aucun physicien expérimental n’a jamais testée pour une simple et bonne raison, il n’y a rien à tester. Il se trouve que tous les tests sur les inégalités de Bell supposent qu’on ait la liberté de choix pour le réglage des détecteurs, et que cette liberté est non-déterministe. Mais on est obligé de choisir entre deux options : ou on utilise un générateur pseudo-aléatoire, ce qui veut dire que la source d’aléatoire n’est pas vraiment aléatoire mais déterministe, et donc on a une corrélation potentielle avec le réglage de l’autre détecteur, ou on suppose ce qu’on désire prouver – que la mécanique quantique est capable de produire de l’aléatoire indéterministe, ce qui fait que les détecteurs sont étrangement intriqués au point de mesure – et donc on change aléatoirement le réglage des détecteurs en positionnant deux générateurs respectifs de nombres aléatoires assez éloignés l’un de l’autre pour être sûr que le choix d’un générateur aléatoire n’influencera pas l’autre.

Donc la question est pourquoi les physiciens insistent-ils sur une interprétation particulière de la mécanique quantique (celle de Copenhague) ? Peut-être parce que l’idée d’un monde déterministe est juste insupportable, nous faisant ressembler à de simples pions sur un échiquier. Selon Kofler “Si l’univers suit un chemin déterministe, alors il n’y a rien qu’on puisse faire en tant qu’expérimentateur” (aussi compatible avec le déterminisme de T Hooft, 2005), compatible avec ce que Bell lui-même envisageait comme une possibilité : si le monde est complètement déterministe, alors même la décision des expérimentateurs de mesurer certaines composantes des spins est entièrement prédéterminée, de sorte que l’observateur ne peut pas avoir décidé de mesurer autre chose que ce qu’il a mesuré de fait. En d’autres mots, les corrélations devraient avoir une explication triviale, car il est trivial de penser à des états corrélés à tous les niveaux, étant donné que tout dans l’univers serait effectivement la conséquence d’un état précédent et partagerait donc simplement une origine commune.

Cependant, si on suppose un monde déterministe on a peu ou même rien à expliquer par rapport aux fondements de la mécanique quantique. On dit souvent que les physiciens quantiques escamotent cette faille, en supposant qu’on ne peut réaliser aucun test. Toutefois, il existe un test possible. Comme Wolfram l’a souligné à de multiples occasions, le pseudo-aléatoire a la particularité évidente de donner exactement le même résultat quand on répète son processus de génération. Il se peut qu’on ait obtenu les mêmes résultats, car on a exécuté les mêmes expérimentations, et que le test lui-même de cette hypothèse pseudo-aléatoire reste encore à faire.

4.11.2 Interprétation informationnelle de la mécanique quantique

Dans le monde macroscopique, l'aléatoire s'accorde visiblement avec la mécanique classique, la situation est fondamentalement différente sous l'interprétation standard de la mécanique quantique. A l'échelle quantique, les choses semblent déconnectées. Quand une particule est intriquée, elle semble perdre véritablement et irréversiblement trace de son état précédent, et il n'y a pas moyen, même en inversant toutes les opérations, d'y revenir. Cela se produit aussi avec la désintégration radioactive, au cours de laquelle un noyau atomique d'un atome instable perd son énergie en émettant des particules ionisées pour lesquelles, selon la théorie quantique, il est impossible de prédire le moment de désintégration, le noyau pouvant se désintégrer n'importe quand avec les mêmes chances.

Toutefois, dans notre monde, l'information ne peut exister que si elle est véhiculée par un processus ; chaque bit doit correspondre à un transporteur physique. Même si ce transporteur n'est pas matériel, prenant alors la forme d'une interaction entre composants de la matière – un atome interagissant avec un autre atome, ou une particule avec une autre. Mais au niveau le plus bas, les plus élémentaires des particules, tout comme les bits isolés, ne transportent aucune information (l'entropie de Shannon d'un bit isolé est 0 car on ne peut pas implémenter un canal de communication de 1 bit seulement, 1 et 0 ayant la même signification s'ils sont isolés, à la fois selon Shannon et pour la complexité algorithmique) quand elle ne sont pas en interaction avec d'autres particules. Elles n'ont pas d'historique causal car elles sont sans mémoire quand elles sont isolées de toute interaction externe. Quand les particules interagissent avec d'autres, elles s'attachent elles-mêmes à un réseau causal et semblent être forcées à définir une valeur en tant que résultat de cette interaction (par ex. une mesure). Ce qui est surprenant pour nous dans le monde quantique, c'est justement ce manque de la causalité qu'on voit pourtant partout ailleurs et auquel on est tellement habitué. Mais c'est l'interaction et son historique causal qui portent toute la mémoire du système dont fait partie le nouveau bit, nous apparaissant comme s'il avait été défini par hasard. Relier un bit au réseau causal peut aussi produire des corrélations de mesures entre des parties de l'espace apparemment déconnectées.

4.11.3 Information quantique et fondements informationnels de la physique moderne

Aucun compte-rendu d'information ou de computation ne peut être considéré comme complet sans avoir pris en compte les interprétations possibles tirées de la mécanique quantique. L'information physique et ses relations avec le calcul réversible sont d'une importance particulière dans la théorie du calcul quantique, et il a été prouvé que si quelque chose peut être calculé de façon réversible, il peut aussi être calculé sur un ordinateur quantique. Bien que cela puisse être plus lent, on a besoin d'un circuit réversible pour tirer avantage de la superposition des qubits, et donc les portes logiques quantiques sont toutes réversibles. Deutsch [Deutsch, 1985] a fourni quelques-unes des bases sur ce sujet, et l'un des plus récents thèmes en développement est le concept de calcul quantique adiabatique.

Selon la mécanique classique, l'aléatoire est apparent dans le monde macroscopique, mais sous l'interprétation standard de la mécanique quantique les choses sont fondamentalement différentes. La position prônant une histoire du monde computationnellement réversible est compatible avec le déterminisme imposé par la mécanique classique. A l'échelle quantique, cependant, les choses semblent différentes. Quand une particule est intriquée, elle semble perdre véritablement et irréversiblement trace de son état précédent, et il n'y a pas moyen, même en inversant toutes les opérations, d'y revenir. Cela se produit aussi avec la désintégration radioactive, au cours de laquelle un noyau atomique d'un atome instable perd son énergie en émettant des particules ionisées pour lesquelles, selon la théorie quantique, il est impossible de prédire le moment de désintégration, le noyau pouvant se désintégrer n'importe quand avec les mêmes chances.

Toutefois, dans notre monde, l'information ne peut exister que si elle est véhiculée par un processus ; chaque bit doit correspondre à un transporteur physique. Même si ce transporteur n'est pas matériel, prenant alors la forme d'une interaction entre composants de la matière (ou de l'énergie) – par ex. une particule interagissant avec une autre. Mais au niveau le plus bas, les plus élémentaires des particules, tout comme les bits isolés, ne transportent aucune information (l'entropie de Shannon d'un bit isolé est 0) quand elle ne sont pas en interaction avec d'autres particules. Elles n'ont pas d'historique causal car elles sont sans mémoire quand elles sont isolées de toute interaction externe. Quand les particules interagissent avec d'autres, elles s'attachent elles-mêmes à un réseau causal et semblent être forcées à définir une valeur en tant que résultat de cette interaction (par ex. une mesure). Ce qui est surprenant pour nous dans le monde quantique, c'est justement ce manque de causalité qu'on

voit pourtant partout ailleurs et auquel on est tellement habitué. Mais c'est l'interaction et son historique causal qui portent toute la mémoire du système dont fait partie le nouveau bit, nous apparaissant comme s'il avait été défini par hasard. Relier un bit au réseau causal peut aussi produire des corrélations de mesures entre des parties de l'espace apparemment déconnectées.

Zeilinger affirme que l'aléatoire quantique est intrinsèquement indéterministe, et que les expériences violant les inégalités de Bell impliquent l'inexistence de certaines propriétés jusqu'à ce qu'elles soient mesurées. Toutefois, ces affirmations sont basées sur une interprétation particulière de la mécanique quantique, à partir de laquelle il tire directement des conclusions en s'appuyant sur divers théorèmes anti-variables-cachées ou d'impossibilité proposés par des gens comme von Neumann, Bell, Kochen et Specker, supposés montrer que l'aléatoire quantique est véritablement indéterministe. Les théoriciens de l'information algorithmique ont deux positions concrètes radicalement différentes, associées à des efforts pour prouver ou infirmer que l'aléatoire quantique est algorithmiquement aléatoire [Zeilinger, 2005, Kochen and Specker, 1975, Von Neumann, 2013].

Ainsi pour Zeilinger, ma position est sur la même ligne que les réponses qui lui sont faites en [Daumer et al., 2006]. Et bien que je croie, avec Daumer et al. [Daumer et al., 2006], que Wheeler n'apporte pas beaucoup d'éclairage sur le problème avec son captivant mais plutôt obscur traitement de l'information en tant que lié, ou même encore plus fondamentalement que cela, à la physique (son dicton "ça vient d'un bit"), je ne partage pas leur point de vue sur ce qui serait faux au sujet de la vision informationnelle du monde. Ils soulignent la remarquable suggestion de Wheeler disant que la physique n'est que de l'information ou que le monde physique lui-même est information. Pour citer Wheeler lui-même [Wheeler, 1990] :

Il n'est pas insensé d'imaginer que l'information repose au cœur de la physique, tout comme elle se tient au cœur d'un ordinateur. Ça vient d'un bit. Pour le présenter autrement, chaque cela – chaque particule, chaque champ de force, même le continuum espace-temps lui-même – tire sa fonction, sa signification, même son entière existence (même si dans certains contextes, c'est indirect) de réponses à des questions oui-non, à des choix binaires, à des bits, arrachés à des appareils.

Cependant dans le même texte, et juste après le passage ci-dessus, Wheeler place une nuance liée à la signification que cela peut avoir quand on introduit un observateur (il faut se rappeler que Wheeler est aussi crédité d'avoir formulé le principe anthropique) :

“Ca vient d’un bit” symbolise l’idée que tout item du monde physique possède en arrière-plan – très profondément, dans la plupart des cas – une source et une explication immatérielle ; ce qu’on appelle réalité survient en dernière analyse, lorsqu’on pose les questions oui/non et qu’on enregistre les réponses sur des équipements ; en bref, l’idée que tout chose physique est théorique-informationnelle à l’origine et qu’on a affaire à un univers participatif.

Ainsi, selon le point de vue de Wheeler, la signification est elle-aussi subjective, ce qui signifie que ce n’est pas un point de vue complètement réductionniste, pas une vision triviale du contenu informatif. La proposition sur laquelle Wheeler insiste le plus, est que “l’information” ne peut se définir en termes de “matière” ou “d’énergie”, et qu’elle peut donc être autant sinon plus fondamentale que la “matière” ou “l’énergie”, les notions les plus basiques en physique. Et une seconde interprétation introduit aussi le problème du contenu informatif, du sens et de l’observateur dans son univers participatif.

En accord avec Wheeler, je pense que le prochain niveau d’unification (à la suite de l’unification d’autres concepts auparavant disjoints en science, tels que électricité et magnétisme, lumière et électromagnétisme, énergie et masse, pour en citer quelques-uns) va concerner l’information et la physique (et finalement, comme une conséquence, la computation et la physique), comme cela a déjà commencé à se produire (par ex. entre la mécanique statistique et la théorie de l’information algorithmique).

Aussi surprenant que cela puisse sembler, c’est l’avis de certains qui sont considérés comme des ontologistes digitaux dans la tradition de Wheeler, Feynman et Wolfram . Dans le cas de Wheeler, son énonciation du principe d’un univers participatif est trop compliquée pour être simplement une position ontologiste digitale. Etant donné qu’il introduit l’observateur et que sa vision possède une forte composante épistémique, il fut probablement plus qu’un ontologiste digital. Feynman se pose le même type de questionnement sur la véritable nature de la mécanique quantique et sur la position “on ne pose pas de questions” (il poursuit en écrivant “Mais cette spéculation, si ce n’est mon propre commentaire, est de même nature que celle faite par d’autres gens – “j’aime”, “je n’aime pas” – et ce n’est pas bon d’avoir des préjugés sur ce genre de choses”). Et l’approche de Wolfram est surtout épistémologique par nature, s’intéressant aux modèles digitaux des phénomènes naturels, même quand le phénomène en question est l’univers lui-même (je fais allusion à sa recherche de la règle ultime, dont il reconnaît qu’elle puisse

ne pas exister tout en pensant qu'il serait dommage de ne pas essayer de la découvrir).

C'est une pratique courante en science que de rechercher les structures plus fondamentales se trouvant en dessous de celles qu'on connaît. Certains chercheurs – dont Wheeler – considérant l'information comme plus fondamentale que la physique elle-même, offrant un support pour finalement donner naissance à l'espace, au temps, à la matière et à toutes les forces connues [Wheeler, 1990, Fredkin and Toffoli, 2002]. D'autres [Feynman et al., 2000], cependant, favorisent une réalité quantique fondamentale comme base pour la description de la réalité, car ils envisagent un univers physique où l'information ne peut faire sens que si elle est portée et incarnée par une entité physique, en l'occurrence des particules quantiques. Landauer écrivait en 1996 (cité d'après Leff et Rex, page 335 [Landauer, 1996, Leff and Rex, 2002]) :

L'information n'est pas une entité abstraite désincarnée ; elle est toujours rattachée à une représentation physique. Elle est représentée par une gravure sur une tablette de pierre, par un spin, par une charge, un trou dans une carte perforée, par une marque sur un article, ou autre équivalent. Cela associe le traitement de l'information à toutes les possibilités et toutes les restrictions de notre monde physique réel, à ses lois physiques et à sa collection de parties disponibles.

Ces auteurs ont développé des visions du monde basées sur la mécanique quantique, et plus spécifiquement sur la computation quantique, où dans les deux cas (qu'ils soient plus fondamentaux ou qu'ils aient un statut ontologique proche de la mécanique quantique), l'information et la computation jouent un rôle majeur. Les tentatives actuelles pour unifier la physique moderne, comme la théorie des cordes ou la gravité quantique avec la propriété holographique, dépendent du codage de l'information et de combien il faut d'information pour décrire quelque chose. Dans ces modèles, c'est l'information qui donne sens aux forces dans l'univers et à la matière elle-même. Ces théories sont supposées introduire une compatibilité entre la Mécanique Quantique et la Relativité Générale grâce au concept d'information (en particulier, l'information maximum), en reliant l'énergie tirée de la théorie quantique et l'énergie tirée de la théorie de la relativité, pour finalement coupler les deux.

Il s'avère que les trous noirs jouent un rôle important, étant donné que l'information maximum, quand les bits correspondent aux photons, est la masse

maximum déterminée par la limite du trou noir. En fait, on peut déterminer maintenant la formation des trous noirs, dans le contexte du principe holographique, comme se produisant lorsque le nombre de bits d'information est égal à la surface du trou noir divisée par quatre et multipliée par l'aire de Planck (10^{-66} m^2). Parmi plusieurs conséquences importantes, l'une d'elle est que le maximum d'information contenu dans un trou noir est fini et que cette information liée à cet objet physique peut se comprendre en termes d'information en deux dimensions (sa surface), raison pour laquelle le principe associé s'appelle holographique – par analogie avec le fait que la surface bidimensionnelle d'un objet contient des informations concernant son volume en trois dimensions [t'Hooft, 2001]. Cette thermodynamique des trous noirs définit la quantité maximum d'information qui peut être stockée dans une région finie donnée de l'espace qui possède une quantité finie d'énergie. C'est assez pertinent à la lumière de l'observation faite par Feynman au cours de l'une des six Messenger Lectures mentionnée auparavant.

Feynman et Bekenstein furent tous deux des étudiants de Wheeler, nous ramenant à la conception informationnelle originale de l'univers (le “it from bit”), qui est aussi compatible avec la vision de Wolfram (lui-même étudiant de Feynman à Caltech) selon lequel on peut définir un espace-temps basé sur des connexions discrètes représentées par exemple par un modèle de réseau trivalent, sur lequel des opérations déterministes simples se produisent au plus bas niveau de réalité, entre les nœuds (contrairement à Fredkin, qui serait plus pour une version automate cellulaire), à partir de laquelle il a été capable d'esquisser une dérivation de la covariance de Lorentz (nécessaire pour la théorie de la relativité). L'idée des réseaux trivalents et leur relation à la gravité quantique a été explorée par Bolognesi dans [Bolognesi, 2001].

Pour résumer, aujourd'hui information et computation jouent un rôle majeur en physique moderne, étant toutes deux à la source de nouvelles théories unificatrices, et suscitant aussi de solides approches sur certains aspects des théories dominantes actuelles telles que la mécanique statistique et la thermodynamique, où elles ont démontré être d'une grande utilité. C'est en effet quelque chose de central pour de nombreux concepts physiques d'aujourd'hui.

Chapitre 5

Calcul : Production d'ordre à partir de l'aléatoire

La contribution de Shannon – en plaçant les informations et les bits dans une relation un-pour-un les unes par rapport aux autres – a contribué à établir un premier pas vers la connexion entre information et physique. Comme l'information est toujours incarnée dans l'état d'un système physique, on peut donc aussi voir la mesure de Shannon comme une quantification des ressources physiques nécessaires pour stocker de l'information classique. D'un autre côté, la théorie de l'information algorithmique formalise et quantifie les concepts de complexité et d'aléatoire d'une façon précise, quoique pas complètement réductionniste, car la mesure reste incalculable bien qu'elle puisse être approchée et applicable.

La théorie algorithmique de l'information a contribué à dévoiler les fondements de la physique, notamment dans le domaine de la thermodynamique, en particulier en aidant à résoudre le paradoxe du Démon de Maxwell [[Feynman et al., 2000](#)]. Les travaux de Landauer et Bennett font la connexion entre physique et information, dans ce qu'on peut voir comme une première tentative de fusion entre ces deux très grands domaines – dans la tradition des fusions de branches auparavant séparées en science. Depuis lors, un important sous-domaine a émergé, la physique de l'information. Deux des articles les plus complets présentant quelques-unes des questions essentielles (pas seulement mathématiques) dans le domaine sont [[Landauer, 1961](#)] et [[Bennett, 1988](#)].

5.1 Information et énergie

La solution du paradoxe du démon de Maxwell a établi un lien direct entre information et entropie physique (thermodynamique). Nous savons maintenant qu'il est impossible de détruire de l'information sans augmenter l'entropie d'un système. Dans un certain sens, l'information peut se concevoir comme interchangeable avec l'énergie. Le principe de Landauer est l'idée que l'effacement de n bits d'information doit toujours avoir pour résultat une augmentation correspondante de chaleur, alors que si aucune information n'est effacée, en principe aucune chaleur n'est produite. Comme Landauer l'a établi [Landauer, 1961], pour qu'un processus computationnel soit physiquement réversible, il doit aussi être logiquement réversible. La limite de von Neumann-Landauer de $kT \log 2$ d'énergie dissipée par bit d'opération irréversible précise que l'énergie finalement émise dans l'environnement est $E \geq kT \log 2$. Il y a eu quelques récents développements dans ce domaine dignes d'être discutés par ailleurs, en particulier de nouvelles objections quant aux solutions informationnelles du paradoxe, qui ont suscité une réponse immédiate de Bennett lui-même et al. pour défendre l'approche de Landauer.

Un résultat important est la connexion entre énergie et logique par le biais de l'information. Le cas des plus basses limites d'énergie thermique relâchée par les différentes portes logiques est quelque chose de différent. Par exemple, la chaleur relâchée par une porte ET est supérieure à celle relâchée par une porte NON, car l'information est détruite dans un ET (les entrées 00, 01 et 10 du circuit binaire sont transformées en une sortie 0, donc à partir d'un simple 0, il n'y a pas moyen de savoir d'où il vient) tandis que le NON est réversible (réversible dans le temps), ce qui signifie qu'on peut obtenir l'information d'origine en appliquant l'opération logique inverse (dans ce cas un autre NON).

La connexion avec l'information, c'est que la perte d'information répand de l'énergie dans la zone environnante sous forme de chaleur. Ce développement est important, car il se trouve qu'on peut bâtir un circuit capable de computation universelle uniquement avec des portes réversibles, comme cela a été prouvé par Fredkin et Toffoli [Fredkin and Toffoli, 2002]. Il existe des combinaisons de portes logiques, tel que l'ensemble constitué de NON et XOR (la disjonction exclusive), avec lesquelles on ne peut pas bâtir de circuit universel. Une porte de Fredkin, par contre, est un circuit computationnel qui permet le calcul réversible en étant capable de supporter toute opération logique ou arithmétique, et donc capable de computation universelle. Ce circuit peut se décrire comme une porte à 3 bits qui échange les deux derniers bits si le premier bit est 1. De même, la porte de Toffoli est une autre confi-

guration de circuit réversible capable de computation universelle. La porte de Toffoli peut aussi être décrite comme une porte à 3 bits qui change le troisième bit si les deux premiers sont égaux à 1.

5.1.1 Computation réversible

Les circuits de Fredkin et Toffoli sont tous deux très simples, réversibles, et peuvent en principe être utilisés pour réaliser des computations optimales sans produire de chaleur ni perdre d'information. Il existe donc un type de computation qui peut être retracée en sens inverse complètement jusqu'à un certain état initial, et comme le suggère Bennett, le monde physique semble correspondre à ce type de computation réversible où information et chaleur peuvent se dissiper mais jamais se perdre. Les recherches sur ces circuits réversibles ont conduit aussi à une étude de leurs relations plus profondes avec le monde physique (sous la forme de processus réversibles en physique). Un modèle idéalisé de calcul basé sur la mécanique classique (newtonienne) appelé un billiard ball computer (ordinateur à boules de billard) fut proposé par Fredkin et Toffoli eux-mêmes [Fredkin and Toffoli, 2002]. L'objectif principal était de formuler une notion de computation basée sur la mécanique newtonienne comme alternative à l'utilisation de signaux électriques comme dans un ordinateur digital. Un des objectifs était d'étudier le problème de l'arrêt dans ce nouveau contexte où aucun algorithme ne peut prévoir si le système arbitraire de boules de billard va s'arrêter, aboutissant à la surprenante conclusion que la question de savoir si un objet mobile peut atteindre une position donnée est indécidable quand il est soumis aux règles de la mécanique classique.

5.2 La question de l'exécution

Si l'univers est algorithmique (c'est-à-dire le résultat d'une computation), on peut supposer qu'il faut un apport de puissance et une programmation. Nous avons proposé l'idée qu'on n'a pas besoin d'injecter de nouvelles informations, ce qui pourrait donner l'impression qu'aucun programmeur n'est nécessaire durant l'exécution du déroulement de l'univers, et a-t-on vraiment besoin d'un programmeur pour faire que l'univers soit algorithmique, c'est-à-dire quelque chose comme un logiciel? Avons-nous besoin d'une certaine sorte d'énergie pour démarrer le programme et ensuite entretenir la computation? Comme Bolognesi le souligne [Bolognesi, 2001], on n'a pas besoin de fournir une réponse, tout comme avec l'autre interprétation de l'univers, la

physique actuelle n’a pas besoin de supposer un calculateur analogique pour expliquer comment les équations d’Einstein ou de Navier-Stokes sont entrées en jeu.

Dans les automates cellulaires, les conditions initiales doivent être mises en place, car c’est nous qui les lançons. Dans la nature, cependant, elles s’appuient simplement sur l’arrière-plan de l’aléatoire apparent du monde (ou si l’on préfère, sur le résultat de cet aléatoire), rendant ainsi beaucoup plus difficile à identifier une computation qui aurait été réalisée. Cela équivaut à démarrer une machine de Turing avec une configuration de ruban non-vierge. Au sens fort, une machine de Turing incorpore une sorte d’énergie sous la forme de la description de l’exécution de ses règles. Car une machine de Turing peut exécuter n’importe quelle opération sous forme de computation, tant qu’on suppose un monde mathématique où on a toujours besoin d’un support pour réaliser les opérations mathématiques. En faisant un ordinateur, on ne devrait plus avoir besoin de séparer les données et les programmes (ou les opérations). Turing a relié les mathématiques pures à la computation d’une façon systématique et universelle. Il se peut qu’on ait encore besoin d’une action initiale d’exécution pour lancer un calculateur, mais une fois lancé, plus besoin d’assistance supplémentaire.

Rolf Landauer a déterminé comment un ordinateur utilise l’énergie lors du traitement de l’information. Il s’avère qu’on peut considérer en principe (sans considérer des raisons secondaires comme par exemple la friction du mouvement qui est extrinsèque au calcul même) que (intrinsèquement) les ordinateurs dépensent de l’énergie seulement quand ils effacent de l’information. On peut ajouter, multiplier ou soustraire des bits sans consommer d’énergie dans l’univers, tant qu’aucune information n’est effacée pendant ce temps. Le principe de Landauer vient de principes physiques robustes. Malheureusement, on doit presque toujours effacer de l’information, pour pratiquement tout. L’effacement émet de la chaleur, qui se dissipe dans l’environnement. En conclusion, l’effacement n’est pas une opération réversible. Effacer n’est pas comme soustraire ou additionner, car soustraction et addition sont des opérations réversibles, tandis qu’effacer entraîne une perte d’information. Même si on peut éviter d’effacer pour calculer, par exemple, Hao Wang a construit une machine de Turing universelle (qu’il appelle machine “B”) qui n’efface jamais [Wang, 1957]. En pratique, effacer est inévitable lorsqu’on écrit, car en écrivant on doit normalement réutiliser des ressources et écrire par dessus (donc effacer) des bits, même si c’est seulement pour faire des additions ou des soustractions.

Nos ordinateurs digitaux peuvent se voir comme le résultat d’une reprogrammation d’une partie de l’univers pour faire en sorte qu’il calcule ce qu’on

veut qu'il calcule. Même si on s'assure de fournir une mémoire propre et neuve à un ordinateur, il va finir par manquer de mémoire dans le monde réel, et va devoir réutiliser sa vieille mémoire (c'est en fait la façon dont Bennett a fini par résoudre le paradoxe du démon de Maxwell – en prouvant que le démon va finir par générer plus de chaleur qu'il n'en économise en choisissant les particules). Cela crée un lien étroit entre mémoire, énergie et entropie (ou dissipation de chaleur) et donc entre computation et thermodynamique. Ce qui montre déjà comment l'information explique les lois physiques de l'univers, et souligne la pertinence qui existe à se poser sérieusement la question de savoir si l'univers peut mieux (ou d'une façon alternative) s'expliquer en ces termes.

Si on veut appliquer la thermodynamique à la mécanique quantique, les choses deviennent plus compliquées car il s'avère qu'il n'existe pas de relations causales pour la plupart, sinon pour toutes les choses qui se produisent au niveau quantique (par ex. pour le temps de désintégration, ou pour la position d'une particule intriquée). Les lois de la thermodynamique peuvent facilement être remises en question dans des situations de la physique (par ex. elles n'expliquent pas le Big Bang ou les trous noirs). Selon l'interprétation standard de la mécanique quantique, par exemple, quand l'état d'une particule s'effondre suite à une intrication, on peut penser du processus qu'il crée une information nouvelle, car rien n'indique quel aurait pu être l'état de la particule avant la mesure qui fut faite. L'état effondré et l'information à son sujet se dissipent alors, s'introduisant dans le réseau causal du monde et amenant un nouveau bit. Que cela soit ou non considéré comme une création d'énergie peut dépendre de la relation entre les entropies de Boltzman et de Shannon.

5.3 Vers un univers constitué de chaînes de bits

Certains disent qu'ils seraient déçus si, parce qu'il serait vu comme discret, on pouvait savoir tout ce qui est connaissable au sujet de l'univers, comme si le caractère discret impliquait immédiatement que l'univers soit trivial. D'un côté ce n'est pas le cas, mais d'un autre côté il se pourrait bien qu'il nous soit largement compréhensible, parce qu'au bout du compte, il est algorithmique.

Donc si on veut savoir si le monde est algorithmique par nature, il faudrait d'abord spécifier ce à quoi pourrait ressembler un monde algorithmique. Si le monde est à tous points de vue un monde algorithmique, les structures

en lui doivent se ressembler, avec des distributions de motifs se ressemblant aussi. Pour montrer cela, nous avons conçu et réalisé [Bennett, 1997] une série d'expériences afin de produire des données par des moyens purement algorithmiques, pour les comparer avec des ensembles de données produites par plusieurs sources physiques.

D'un côté, on a pris des échantillons tirés de sources physiques. Au bon niveau, les données peuvent toujours s'écrire en binaire, car chaque observation physique conduisant à des valeurs (poids, position, etc..) peut s'énumérer indépendamment sous forme de séquence discrète. Chaque distribution de fréquences est le résultat du comptage du nombre d'occurrences de k -tuples (sous-chaînes de longueur k) extraites à partir des chaînes binaires venant à la fois des ensembles de données empiriques et à la fois de données digitales obtenues d'après les résultats de machines digitales. Donc d'un autre côté, on produit une version expérimentale de $m(s)$ en faisant tourner un large ensemble de petites machines de Turing pour lesquelles le temps d'arrêt est connu (grâce au problème du Castor Affairé). On a ensuite comparé statistiquement les fréquences de distributions produites (voir [Kolmogorov, 1965]). On effectue un test de niveau de corrélation et sa signification est mesurée pour valider soit l'hypothèse nulle, soit l'hypothèse alternative (l'hypothèse nulle étant que les ressemblances entre données empiriques et monde digital sont dues au hasard).

La théorie générale de Solomonoff sur l'induction débute par un théorème crucial qui fut publié pour la première fois en 1960 comme faisant partie de son invention de la probabilité algorithmique. Le théorème déclare que parmi les algorithmes qui décodent les chaînes à partir de leurs descriptions (codes), il en existe un optimal. Pour toutes les chaînes, cet algorithme autorise l'utilisation de codes aussi courts que ceux permis par n'importe quel autre algorithme, additionnés d'une constante qui dépend des algorithmes mais pas des chaînes elles-mêmes. Solomonoff a utilisé cet algorithme et les longueurs de code qu'il permet, pour définir une probabilité algorithmique de chaîne, sur laquelle on peut se baser pour introduire une déduction sur les prochains digits d'une chaîne.

5.4 Un monde digital simulé

On a trouvé certaines ressemblances et différences entre le monde physique et le monde purement algorithmique. Pour citer une différence typique, dans l'étude des groupes de symétrie pour les chaînes, les chaînes du type $(01)^n$ (c'est-à-dire 01 n fois) adoptent un niveau bas dans les distributions

de données empiriques, contrairement aux données digitales, qui adoptent un niveau plus élevé. Tout comme les processus ne démarrent pas à partir d'un ruban vierge (pour utiliser les machines de Turing comme analogie), dans le monde réel, les chaînes ne sont pas délimitées habituellement ; on ne peut pas vraiment dire quand un phénomène physique s'est arrêté, seulement quand les mesures sont achevées. Dans le monde réel, la probabilité de détruire des chaînes hautement symétriques à cause de ce manque de délimitation est plus haute que lorsqu'on assemble une chaîne symétrique en changeant ses bits au hasard. Il n'y a pas moyen de dire quand un processus commence ou s'arrête dans la nature, et de même pas moyen de mettre en place des mesures en prenant en compte les "bonnes" longueurs de temps des flux de données empiriques.

Dans le monde simulé, les choses sont différentes car les machines possèdent une configuration d'arrêt, ainsi qu'un état spécial d'arrêt. Et parce que dans notre monde simulé, les machines n'ont pas d'interaction entre elles, les chaînes périodiques et hautement symétriques ont de plus grandes chances de rester intactes, adoptant un rang supérieur dans la fréquence des chaînes disposées de la plus haute à la plus basse fréquence (donc de la plus basse à la plus haute complexité aléatoire). Pour vérifier que c'est vraiment le cas, nous avons monté deux autres expériences différentes. L'une consistait à lancer des machines à partir de conditions initiales aléatoires. Même si cela ne simulait pas complètement l'interaction entre processus à toutes les étapes, c'était un montage plus proche de ce qui se passe dans le monde réel, où des computations commencent habituellement quand d'autres prennent fin. L'autre expérience a consisté à lancer des machines non auto-limitées comme les automates cellulaires à une dimension, qui n'ont par définition aucune configuration d'arrêt. Leur computation peut s'arrêter n'importe quand, donnant des chaînes de longueurs arbitraires, tout comme cela se passerait dans la plupart des expériences dans le monde réel, où on doit classiquement décider quand arrêter de faire des mesures pour commencer une analyse de fréquences. On a trouvé ce à quoi on s'attendait : les chaînes $(01)^n$ hautement structurées sont descendues dans le classement par rapport à la première expérience avec les machines dotées d'arrêts, et sont plus proches du niveau de distribution des données empiriques.

Ce qui se passe, c'est que dans le monde réel, les chaînes hautement organisées ont peu de chances de se constituer si elles interagissent avec d'autres systèmes. Changer un simple bit détruit le motif parfaitement périodique d'une chaîne $(01)^n$, et plus la chaîne est longue plus grandes sont les chances qu'elle soit détruite par une interaction avec un autre système ou parce qu'elle a été coupée au mauvais endroit à l'instant de la mesure. Dans le cas des ma-

chines de Turing qui s'arrêtent, cependant, les chaînes sont délimitées par le code disant à la machine de produire exactement n alternances et de s'arrêter.

Si la probabilité de changer n'importe quel digit dans une chaîne s est uniformément distribuée (tout digit ayant la même probabilité d'être modifié), il s'en suit que $Pr(010101 \rightarrow 101010) > Pr(000100 \rightarrow 001000)$, où la flèche signifie que la chaîne s_1 devient s_2 . Ou alors en mesurant la distance de Hamming, il s'avère que le nombre de modifications qu'une chaîne a besoin de subir pour rester dans la même classe de complexité peut se décrire comment suit : $HammingDistance(010101, 101010) = 6$, $HammingDistance(000100, 001000) = 2$. En d'autres mots, le plus court chemin pour transformer 010101 en 101010 requiert six changements pour préserver sa complexité, tandis que la chaîne 000100 en requiert deux pour devenir 001000, qui est une chaîne de la même classe de complexité. Il est clair que la probabilité d'apparition de six changements précis de bit est plus basse que la probabilité de deux changements de ce type. De plus la seule chance pour que la première chaîne 010101 reste dans la même classe de complexité est de devenir la chaîne particulière 101010, tandis que pour la seconde chaîne 000100, il y a d'autres possibilités : 001000, 110111, 111011.

Les calculs réalisés ont fourni assez de données pour produire un $m(s)$ empirique ayant des chaînes jusqu'à relativement courtes (à partir de petites machines de Turing ayant jusqu'à 4 états), pour lesquelles on a pu évaluer $C(s)$, une chose qui n'avait jamais été faite auparavant à cause de la difficulté, étant donné que le moyen habituel d'approcher $C(s)$ est par les algorithmes de compression. Pour de courtes chaînes, cela ne marche pas pour des raisons évidentes (les algorithmes de compression n'arrivent pas à trouver de structures dans les chaînes trop courtes, donc les valeurs issues des compresseurs sont trop instables). En fait, on appelle aussi $m(s)$ la distribution universelle, car on n'y suppose rien d'autre que l'utilisation d'une machine universelle de Turing. Ainsi (comme cela fut prouvé par Levin lui-même [Zvonkin and Levin., 1970]), cette distribution domine toute autre distribution demi-calculable (on l'a aussi appelée la distribution miraculeuse [Kirchherr et al., 1997]). Si on voulait créer, par exemple, des candidats-univers, il faudrait probablement d'abord être sûr qu'un univers soit capable d'approcher le $m(s)$ empiriquement calculé, ce qui serait une indication qu'il est capable de produire assez de complexité à la fois en termes de complexité structurée et d'aléatoire apparent distribués comme $m(s)$ le spécifie, et ensuite on pourrait regarder si l'univers satisfait toutes les autres lois physiques (comme l'invariant de Lorentz).

Néanmoins, en introduisant la probabilité d'une chaîne s , l'intérêt centré sur l'espace S des chaînes auquel s appartient, se reporte sur l'espace P des

programmes pouvant calculer s . Mais alors, pourquoi ne pas supposer que même les éléments de l'espace P – les chaînes elles-mêmes – jouissent d'une probabilité a priori ?

La beauté de $m(s)$ vient du fait que peu importe, à la base, d'où (ou de quoi) on part, on finit par avoir la même distribution car ce qui importe, c'est le filtre computationnel, la distribution aléatoire des programmes. Je pense que seule la distribution des programmes aurait un impact sur $m(s)$ (nos expériences le confirment aussi). Une question intéressante qui surgit est si l'on peut effectivement composer des restrictions sur la distribution des programmes, par exemple des restrictions fondées sur des lois physiques qu'on pourrait modéliser avec la théorie des jeux (quelque chose qui aurait une affinité avec les critiques de Kevin Kelly au sujet des approches bayésiennes du vieux problème de l'induction).

Mais en fait, suite à notre recherche [Delahaye and Zenil, 2012], $m(s)$ n'est plus un a priori. Notre $m(s)$ expérimental (avec toutes les réserves dues à sa portée limitée) n'est plus une distribution bayésienne mais empirique (donc postérieure), qui, selon Kevin Kelly, donnerait à notre approche, ainsi qu'au paradigme de la complexité algorithmique, une plus grande légitimité théorique. On peut voir de la complexité émerger de nos distributions expérimentales en accord avec la façon dont $m(s)$ et $C(s)$ sont supposés opérer.

5.5 L'ADN est simple, mais le déploiement des protéines est complexe

La biologie a été témoin d'une transformation durant le siècle dernier, en commençant par les découvertes de Mendel sur le transfert de certains caractères dans les plantes (petits pois), un phénomène équivalent au transfert d'information entre générations. Redécouvertes sur le tard, ses lois auraient posé les fondements de ce qui est aujourd'hui la génétique moderne, qui établit que les organismes vivants stockent et lèguent des informations comprenant des instructions pour leur développement et leur fonctionnement, encodées dans les acides ribonucléiques et nucléiques. Pour illustrer comment les processus naturels se comportent algorithmiquement, et pour empiriquement spécifier ce que cela signifie concrètement, le cas de l'ADN est un parfait exemple, associant l'information et le monde physique (et biologique).

Parmi les processus connus pour être impliqués dans la réplication et la transformation de l'ADN, il y a la translocation des chromosomes (un fragment d'un chromosome est détaché pour ensuite être attaché sur un autre

chromosome), la transcription réverse, l'échange et le croisement de fragments de code, le rapprochement (appariements par liaisons hydrogènes à une séquence complémentaire) et la dénaturation de l'ADN (séparation de longueurs à simple brin en cassant les liaisons hydrogènes entre les bases). Ce sont tous des processus relativement simples. Un sous-ensemble d'opérations purement digitales peut faire s'accorder ces opérations avec des computations. Des opérations comme le jointage, copiage, partitionnement, complémentation, découpage ou remplacement sont équivalentes à celles qu'on peut voir dans l'ADN.

Les règles déterminant la façon dont l'ADN se réplique peuvent aussi être du même type algorithmique que celles pilotant d'autres types de phénomènes physiques, nous conduisant parfois à découvrir de fortes ressemblances dans leur distribution de groupons (tuples). La fabrication de l'ADN, à part peut-être l'opération de mutation (le résultat d'une interaction avec un autre système n'est pas nécessairement une véritable opération aléatoire) est le résultat d'une longue période d'application de règles simples, couches après couches de codage du vivant, développées sur des millions d'années suivant un profond processus algorithmique avec ses propres règles caractéristiques, rendant des processus, tel que le déploiement des protéines, comme hautement complexes à nos yeux.

Mais c'est une chose que d'ignorer l'algorithme du déploiement des protéines, et autre chose que de prétendre que ce déploiement ne peut pas être supporté par une machinerie (déterministe, en temps polynomial ou non, on ne sait pas encore). Ne pas connaître l'algorithme de déploiement des protéines ne suffit pas pour en déduire qu'il n'en existe pas.

Dans le cas de structures dont on est certain de l'état final (que ce soit une protéine déployée ou la division d'une cellule), la solution est soit le résultat d'un algorithme, soit un processus aléatoire. Il est improbable que le déploiement des protéines soit aléatoire car, comme on me le fit remarquer, une protéine incorrectement déployée devrait être cause de maladie. Mais étant donnée la robustesse de la nature dans son entier, un déploiement aléatoire à gauche au lieu d'à droite ne devrait pas faire de différence pour le système. Cependant, comme le système est sensiblement dépendant de la configuration correcte, on pense que le sous-système est lui aussi dépendant du retour d'information venant du système.

Si la computation est la force conductrice produisant de la structure dans le monde, on peut l'utiliser comme base pour influencer l'orientation de la complexité. Je pense au problème de la soupe primitive, réunissant les bons éléments chimiques pour l'apparition de la vie. Se pourrait-il que cela échoue

parce qu'on n'a pas inclus dans le mélange le type de computation qui produit la force de complexification ? Si on s'attend à ce que la vie émerge à partir de produits chimiques sans ajouter de processus d'activation, cela peut équivaleoir à attendre que les produits chimiques se mettent à écrire directement les décimales de π , plutôt que le programme produisant π . Comment la nature fabrique-t-elle de tels programmes ? Wolfram, par exemple, pense que la nature explore ce qu'il appelle l'univers computationnel des programmes possibles.

Comme Wolfram l'affirme, le concept d'évolution darwinienne peut nous laisser supposer que tous les processus donnant aujourd'hui naissance aux formes que nous voyons dans les systèmes biologiques doivent avoir été soigneusement façonnés par la sélection naturelle. Wolfram soutient [Wolfram, 2002], en l'occurrence grâce aux motifs ornant les coquillages, que la nature échantillonne les programmes au hasard.

Tout le monde n'est pas d'accord sur le formalisme permettant de décrire le processus d'émergence, comme dans le cas où un motif émerge à la surface d'un coquillage. Mais la plupart des chercheurs pensent que c'est une propriété de quelque système complexe composé de multiples éléments en interaction, partant d'une dynamique relativement simple au niveau des entités individuelles composant le système.

De plus les motifs ne se répètent pas en se copiant eux-mêmes, ils se répètent car ils sont le résultat d'un programme simple produisant le même motif. Si le motif change, cela signifie que la règle a aussi changé. La sélection n'a pas lieu entre plusieurs nouveaux motifs possibles mais plutôt entre programmes simples possibles. Pour étayer cette position, considérons le nombre de motifs qu'il est possible de sélectionner. Si on était libre de sélectionner les motifs, le résultat serait plutôt aléatoire, car on serait capable de changer de motifs n'importe quand et de n'importe quelle façon. Or ce n'est pas ce qui se passe, c'est pourquoi on ne voit pas les coquillages modifier leurs motifs en plein milieu de leur coquille. Dit autrement, leurs motifs sont robustes car leur processus de formation suit une règle et ce n'est pas une simple affaire de mimétique à partir de motifs déjà existants. Les motifs sont robustes car ils sont le résultat d'une petite règle concise.

5.6 La vie est robuste car elle suit des règles

Il est reconnu qu'on peut tracer une route entre la soupe primitive et le monde de l'ARN, et que cela aide à expliquer comment on obtient l'ADN et

les protéines. On sait que le monde de l'ARN est la clé pour comprendre la vie. La question est comment le saisir en tant qu'étape intermédiaire de la vie et donc comment l'utiliser pour expliquer la vie. On sait que le monde de l'ARN fut une étape primitive dans l'évolution de la vie.

L'expérience de Miller, qui a produit avec succès certains des éléments organiques de base de ce qu'on croit avoir été l'atmosphère primitive terrestre, s'est révélée assez robuste, au sens où les types d'éléments produits ne dépendent pas beaucoup des paramètres choisis. Si c'est la façon dont la vie a commencé sur Terre, pourquoi l'expérimentation de Miller n'a-t-elle pas donné naissance à un ours ou un singe ? Bien sûr, chaque expérimentation a ses limites, tout comme a ses limites chaque plan à chaque étape de la vie. Cela peut s'expliquer en partie par la coexistence de différentes méthodes et causes, parmi lesquelles il peut y avoir de l'information basée sur la dynamique d'un processus. Michel Morange [Morange, 2012], par exemple, défend l'idée que la vie ne peut pas s'expliquer comme le résultat d'une force unique, et qu'à côté de l'évolution darwinienne d'autres explications mécanistes et physiques peuvent exister pour rendre compte de son origine et de sa propre évolution. La clé de la vie peut, après tout, s'avérer être très simple. Il se peut que nous ne comprenions pas la transition entre chimie et biologie parce que, justement, elle n'a rien de spécial, tout comme n'a rien de spécial le passage de l'eau liquide à l'état solide à une certaine température. La vision dogmatique que l'ADN est particulièrement spécial est peut-être une vision moderne du monde centré sur l'ADN, comparable aux visions uni-centrées précédentes d'un type ou d'un autre. Si la question est pourquoi on ne voit pas de vie émerger quand on met certains éléments chimiques bruts dans un tube à essai, alors il se pourrait bien que ce ne soit plus la question à poser, car je pense qu'on est en train d'observer cela précisément dans les récents développements de la biologie synthétique. Pourquoi l'ADN ou la vie demanderaient-ils une explication complètement différente par rapport au reste de l'univers ? Il existe certains processus dans l'univers pour lesquels il ne serait pas possible de dire s'ils sont supportés par quelque chose de vivant ou non, s'ils ont l'air vivants ou non pour un observateur. Le Jeu de la Vie de Conway [Gardner, 1970], par exemple, (conçu pour exploiter précisément des caractéristiques du vivant) montre de nombreuses propriétés habituellement assignées au vivant. Le Jeu de la Vie n'est nullement un cas isolé de ce genre de comportement ; c'en est un parmi de nombreux systèmes capables de mêmes comportements compliqués. Néanmoins les choses qui s'auto-répliquent sur Terre semblent toutes être vivantes et basées sur un même langage et plan organisationnel sous forme d'ADN. Il se peut que le non-basé sur de l'ADN nous paraisse non-vivant. Wolfram [Wolfram, 2002]

suggère que notre identification des formes de vie tient plus d'un cheminement historique commun que nous partageons avec toutes les espèces sur Terre, que de quelque chose de vraiment plus spécial.

La sélection naturelle fonctionne bien pour les systèmes qui ont déjà évolué vers un système héritable de réplication. Darwin savait (et l'avait admis) que sa théorie ne pouvait pas expliquer l'origine de la vie. La sélection naturelle intervient après que la vie ait émergé. Lors de ses expérimentations informatiques de 1953, Nils Barricelli en avait déjà conclu que "mutation et sélection seules ... [se sont montrées] insuffisantes pour expliquer les phénomènes d'évolution" [Barricelli, 1963]. Les ingénieurs de Princeton en charge de faire tourner les expériences de Barricelli sur l'émergence de la vie l'ont interrogé, en croyant qu'il y avait quelque chose dans son code au sujet duquel ils n'avaient pas été mis dans le secret. George Dyson [Dyson, 1998] interprète cela comme une réflexion sur la surprise des ingénieurs voyant qu'un ensemble d'instructions simples pouvait générer une complexité croissante.

Tandis que la sélection naturelle a été étudiée sous toutes les coutures depuis Darwin, le processus de *complexification* (qui est la force motrice produisant les structures organisées à partir de l'aléatoire) était habituellement affirmé ou supposé. La *complexification* (j'ai choisi ce terme pour éviter les galvaudés "auto-organisation" et "émergence") est le processus par lequel une structure ou un motif apparaît dans un système en l'absence d'autorité centralisée ou de dirigisme extérieur. On pense que les motifs viennent des interactions locales des éléments qui constituent le système. Maintenant si un système est computationnel, m ne nous dit pas seulement comment les motifs vont être distribués mais aussi comment la structure émerge du système lui-même, sans autorité centrale et sans direction de l'extérieur, mais néanmoins d'une façon très précise et inévitable à la base. Un exemple de complexification est la différenciation locale dans les systèmes biologiques. La complexification implique l'idée de computation. La succession des générations n'est qu'une partie du déploiement de ces systèmes, et l'héritage peut s'interpréter comme un transfert d'information.

La vie, en tant qu'intelligence, est probablement l'une des plus, sinon la plus étonnante organisation de la matière, mais il me semble qu'il y a seulement une différence de degré entre vie et non-vie, et qu'il ne se passe rien, dans les coulisses, d'essentiellement différent de ce qu'on a déjà expliqué. Avec la météo, par exemple, on a déjà abandonné les tentatives pour en comprendre plus. On suppose naturellement aujourd'hui qu'il n'y a rien de spécial derrière le temps qu'il fait, c'est juste simplement que nous ne le comprenons pas assez bien pour faire des prévisions précises à long terme, c'est-à-dire pour plus de quelques jours et parfois le jour même. L'explication

ci-dessus des systèmes computationnels vis-à-vis de la force de “complexification” peut paraître spéculative, et jusqu’à un certain point elle l’est, mais le modèle fonctionne vraiment bien et nous sommes en position pour expliquer le processus expérimentalement comme on l’a montré (en produisant une distribution à partir de faire fonctionner des milliers de machines de Turing petites). En plus de l’objet technique de nos travaux, nous démontrons comment produire de la complexité organisée à partir de rien (ou à partir de l’aléatoire), et pas seulement sur le papier mais en déployant une machine universelle, qui, sans autre apport, sauvegarde ce que donnent des programmes aléatoires lancés sur elle. En d’autres termes, comme on l’a vu ci-dessus, le processus de complexification est pratiquement inévitable, sauf pour des machines dégénérées trafiquées à la main (qui, cependant, ne peuvent pas se comporter ainsi bizarrement pendant longtemps). Une fois qu’on démarre une machine universelle naturelle sur des programmes aléatoires (se souvenir que Turing a démontré que programmes et données sont interchangeables), on finit avec une structure organisée très spécifique liée à la demi-mesure de Levin .

5.7 Les données empiriques sont compressibles

L’affirmation que les données empiriques, peuvent se traiter comme une distribution (c’est-à-dire la fréquence de certains motifs par rapport à d’autres), comme si les données empiriques étaient toutes de même nature, doit se faire évidemment avec grande prudence. Il faudrait d’abord montrer qu’il existe une distribution générale collective derrière la plupart/toutes sortes de données empiriques. C’est en effet quelque chose que nous avons testé et rapporté dans [Delahaye and Zenil, 2011]. La preuve que la plupart des données empiriques transportent un fort signal algorithmique est que la plupart des données sont compressibles à un degré plus ou moins grand. Pensons aux divers types de données que nous stockons dans votre ordinateur, que ce soit de la musique, des images ou du texte, tout ceci hautement (sans pertes) compressible.

Une donnée empirique encode une succession de symboles ou d’évènements. Un ensemble de données empiriques comme l’ADN, par exemple, utilise un alphabet de 4 bases – A, C, G et T. Il faut 2 bits pour représenter 4 symboles. Les algorithmes spécialisés de compression sont capables de compresser les séquences d’ADN sous les 2 bits par base, en allant jusqu’à environ

1,6 bits par base. D'après la complexité algorithmique, on sait qu'un processus n'est pas aléatoire quand il est algorithmiquement compressible. Les chercheurs compressent l'ADN car la compression équivaut à la détection de motifs ; l'ADN codant certains gènes possède des taux de compression différents qui aident à identifier les positions où certains segments apparaissent. Cependant au moins un chercheur a mis en doute cette idée [Griffith et al., 2005], quoiqu'il n'ait pas réussi à en convaincre d'autres, car les données (ou leurs propriétés distinctes) semblent effectivement bien être hautement compressibles [Twardy and Gardner, 2005, Sato et al., 2001].

McAllister caractérise les données empiriques de la façon suivante [McAllister, 2003] : "Avant d'être appréciées pour leur valeur probante, leur valeur de garantie empirique, les données empiriques sont habituellement soumises à différentes formes de traitement, par exemple pour corriger ou compenser des facteurs supposés avoir affecté les observations ou les mesures. Ce que j'appelle données empiriques est un ensemble de résultats observationnels avant un tel traitement." Selon Mc Allister, les données empiriques qui sont produites sous la forme d'une somme de deux composants, un signal compressible et un bruit incompressible, sont compressibles. Toutefois, grâce à un exemple codant les résultats d'un jet de dé truqué en utilisant un codage par mot de passe qui prend en compte le trucage sous la forme d'une distribution de probabilité, Twardy et al. [Twardy and Gardner, 2005] montrent que la Thèse de la Sommutation est fausse.

On pourrait se demander si la compressibilité des données est, en un sens, une indication de la nature algorithmique du monde. D'un côté, on a le problème de la compression sans pertes, car il n'y a pas moyen de prouver que les dépôts digitaux emmagasinés préservent toutes les données, même si la source n'est pas du type digital. Au niveau le plus petit et le plus basique, si le monde est discret, il peut sembler que la compression sans pertes de n'importe quelle donnée empirique soit possible, mais les méthodes traditionnelles sont basées sur le concept d'échantillonnage, qui en pratique est suffisant pour saisir la plupart des caractéristiques distinguables des données dans le monde réel, et on peut toujours en saisir plus ou moins selon les usages pour lesquels les données stockées sont destinées. Dit autrement, plus c'est précis, plus la fréquence d'échantillonnage est élevée. D'un autre côté, le fait que la donnée puisse être compressée à la fois sous forme de dépôts analogiques et digitaux, est une indication de régularités présentes dans la donnée, et une description algorithmique de ces régularités.

Si le monde était aléatoire, la plupart des choses seraient algorithmiquement aléatoires simplement parce que le nombre des descriptions courtes est toujours plus petit comparé au nombre de possibilités combinatoires d'un ob-

jet. Par exemple, le nombre de descriptions binaires de toutes les chaînes de bits de longueur n moins 10 digits est 2^{n-10} . Si n est grand, seules quelques-unes seront compressibles de 10 digits, mais plus la chaîne est courte, moins il y a de descriptions. En fait, la plupart des chaînes vont avoir des complexités aléatoires maximum. Mais ce n'est pas le cas quand on utilise les algorithmes de compression sur des données empiriques. Si les données empiriques étaient de ce type probabiliste, elles seraient hautement incompressibles, mais elles ne sont pas de ce type. Les données empiriques sont hautement compressibles, parce qu'elles ne sont pas produites au hasard mais par des processus spécifiques. Le texte par exemple, suit ses propres règles, répète certaines lettres et combinaisons de lettres régulièrement (en ne dépendant que de la langue). Donc en compressant du texte, on est capable de réduire sa taille fortement. C'est le cas avec la plupart des données empiriques, mêmes les données produites par la nature. Pensons au soleil, par exemple – si on avait encodé le lever et le coucher du soleil avec 0 et 1. Les chances de trouver des données incompressibles dans un monde analogique pourraient toutefois être bien plus grandes, simplement parce que les possibilités pour toutes choses sont bien plus grandes.

Il peut encore sembler nécessaire d'expliquer pourquoi les théories du continuum fonctionnent si bien aux échelles qu'on peut tester. Peut-être que c'est simplement parce que les théories discrètes ne peuvent fonctionner qu'aux échelles qui rendent la tâche computationnelle de la prévision intolérable. Dans ce cas, il serait tentant d'exclure les modèles discrets en tant que non-pertinents d'un point de vue pratique. Mais les modèles discrets se sont avérés réussir incroyablement bien, non seulement pour décrire le monde, mais aussi pour trouver des applications et faire des prévisions.

Pour finaliser l'analyse, la différence essentielle entre un monde analogique et un monde digital, c'est la forme concrète des infinis. Un monde analogique voudrait dire qu'il y a une infinité de parties, que ces parties sont continuellement connectées, ou qu'il n'y a aucune partie et que la matière et tout le reste reposent de façon lisse dans un espace physique continu, en principe capable de suivre n'importe lequel parmi l'infini des chemins et trajectoires possibles. Il peut sembler possible de donner grand sens à cela au moyen des différents domaines de calcul différentiel et d'analyse mathématique, pourtant il faut faire une distinction entre ce que les symboles représentent et ce que sont les calculs concrets parmi les symboles.

Si le monde est analogique, il aurait plus de chances de ressembler à l' Ω de Chaitin, un nombre aléatoire par définition, dépendant de l'imprévisibilité des calculateurs digitaux universels, un nombre appelé aussi probabilité d'arrêt. Notre monde pourrait être un petit fragment, apparemment ordonné, au

milieu d'un univers algorithmiquement aléatoire, comme Calude, Salomaa et Meyerstein le suggèrent [Calude and Meyerstein, 1999]. Auquel cas, on devrait se considérer comme très chanceux de vivre dans ce minuscule îlot apparemment ordonné, comme ils le soulignent.

5.8 Révéler la machinerie

Un processus physique peut se voir comme une computation. Les contraintes sont une raison pour compliquer une théorie, par exemple des contraintes physiques ou des contraintes du style théorie des jeux, qui peuvent s'ajouter au modèle algorithmique minimal. Mais si on n'a aucune raison de choisir une contrainte spécifique, et qu'aucune information a priori n'est disponible sur l'ensemble des données, la distribution uniforme est celle qui ne fait aucune supposition, selon le principe d'indifférence. Considérons une opération inconnue générant une chaîne binaire de longueur k bits. Si la méthode est uniformément aléatoire, la probabilité de trouver une chaîne particulière s est exactement 2^{-k} , la même pour toutes les chaînes de longueur k , équivalent aux chances de tomber sur les décimales de π .

Cependant les données (tout comme π – largement présent, par exemple, dans la nature sous la forme de processus courants liés aux courbes) sont habituellement produites non pas par hasard mais selon des processus spécifiques. Il existe une mesure qui décrit la forme attendue de la distribution de fréquence d'une machine abstraite tournant avec un programme aléatoire. Un processus qui produit une chaîne s avec un programme p quand il est exécuté sur une machine de Turing universelle T possède la probabilité $m(s)$ [Zvonkin and Levin., 1970]. Pour toute chaîne donnée s , il y a un nombre infini de programmes pouvant donner s , mais $m(s)$ est défini de façon à ce qu'on puisse assigner une probabilité à une chaîne étant produite par un programme aléatoire.

L' Ω de Chaitin et $m(s)$ sont complètement liés. En fait $m(s)$ contient Ω . Tandis que l' Ω de Chaitin est la probabilité d'arrêt, $m(s)$ est la probabilité de sortie (des programmes s'arrêtant lancés sur une machine de Turing universelle à préfixe libre), donc en connaissant Ω , on a $m(s)$. La distribution $m(s)$ a une autre particularité intéressante : on peut partir de presque n'importe quoi, et avec la plupart des distributions probabilistes, elle reste largement inchangée. C'est le processus qui détermine la forme de la distribution, et pas les conditions initiales d'où part le programme. C'est important car on ne fait pas de suppositions quant à la distribution des conditions initiales mais plutôt sur la distribution des programmes. Des programmes tournant

sur une machine de Turing universelle devraient être uniformes, ce qui ne veut pas nécessairement dire véritablement aléatoires. Par exemple, pour approcher $m(s)$ par en dessous, on peut concrètement définir un ensemble de programmes d'une certaine taille et définir une énumération pour lancer systématiquement les programmes un à la fois.

C'est important car cela signifie que du véritable aléatoire, le type d'aléatoire supposé dans la mécanique quantique, n'est pas vraiment nécessaire. Ainsi on n'a pas réellement besoin de la mécanique quantique pour expliquer la complexité du monde ; pour expliquer la réalité, on n'a pas besoin d'établir que la mécanique quantique la soutient.

Tout comme pour les chaînes qui peuvent être produites par des programmes, on peut interroger la probabilité d'un certain résultat tiré d'un certain phénomène naturel, pour savoir si le phénomène, tout comme pour une machine calculante, est le résultat d'un processus plutôt que d'un événement aléatoire. Si on ne dispose d'aucune autre information sur le phénomène, on peut voir si $m(s)$ dit quelque chose sur une distribution des résultats possibles dans le monde réel. Dans un monde de processus calculables, $m(s)$ indiquerait la probabilité que possède un phénomène naturel pour produire un résultat particulier et indiquerait la fréquence d'une certaine structure. Si on devait parier sur certains événements dans le monde réel sans avoir aucune autre information, $m(s)$ serait une décision raisonnable si le monde était digital, tout comme l'est $m(s)$ pour des machines abstraites (digitales).

La distribution $m(s)$ de Levin fournit la distribution a priori d'une chaîne binaire s , et dépend du nombre de programmes de n'importe quelle longueur qui déclenchent une computation sur une machine de Turing universelle à préfixe libre se terminant par la production de s . Ainsi la somme des $m(s)$ de tous les s dépend du nombre de programmes déclenchant une computation sur une machine de Turing universelle à préfixe libre qui s'arrêtent (en produisant un s quelconque), et c'est l' Ω de Chaitin. $m(s)$ est au plus bas, demi-calculable, ce qui signifie qu'on peut l'approximer (avec beaucoup de ressources computationnelles), mais jamais le calculer complètement (à cause du problème de l'arrêt). Mais les temps d'activité avant arrêt sont connus pour de (relativement) grands espaces de machines abstraites, par exemple pour les machines de Turing grâce au jeu du Castor Affairé. Donc une des tâches qu'on a entrepris fut de calculer un $m(s)$ expérimental pour les valeurs des fonctions Castor Affairé disponibles et connues. Un important corollaire vint du fait qu'on a choisi un moyen (peu importe sa limitation) de calculer $m(s)$, puis un moyen de calculer $C(s)$, la complexité algorithmique de la chaîne s , grâce au théorème du codage de Chaitin-Levin . Et cela constitue l'applicabilité de notre recherche au delà des comparaisons statistiques entre

les ensembles de données du monde réel et les ensembles de données artificielles/digitales (pour tester l’hypothèse computationnelle) comme reporté dans [Joosten et al., 2011].

Il n’y avait jusqu’ici aucun autre moyen d’aller au-delà d’une simple supposition d’un univers calculateur, en confirmant ou réfutant l’hypothèse algorithmique, c’est-à-dire en déterminant si vraiment l’univers pouvait se concevoir comme le résultat d’un certain programme de calcul ou alors d’une certaine sorte de vaste appareillage de computation digitale, comme cela a été suggéré par quelques auteurs [Fredkin and Toffoli, 2002, Wolfram, 2002, Zuse, 1969]. Nous avons cependant été capables de mesurer jusqu’où les distributions d’ensembles de données (chaînes binaires) tirées de notre monde réel ou tirées d’un monde artificiel algorithmique, approchaient une distribution empirique $m(s)$. Nous avons trouvé que, tandis que la corrélation entre données réelles et données digitales n’étaient pas assez significatives pour être source de conclusions, chaque distribution était corrélée – selon différents degrés de confiance – avec au moins une distribution algorithmique produite par un modèle de computation. En d’autres mots, les distributions tirées de données empiriques dispersent les structures de la même façon que les distributions générées en se servant de calculateurs digitaux (machines de Turing, automates cellulaires et autres machines abstraites). Dans un univers algorithmique, on finit avec un monde organisé en suivant une distribution très spécifique (la distribution universelle de Levin).

Notre travail [Joosten et al., 2011] est une justification de la position “singé tapant au hasard sur une machine et qui peut produire de l’organisation s’il est devant un ordinateur plutôt que devant une machine à écrire” comme elle a été formalisée par Levin (et, d’une autre façon, par Solomonoff) dans ce qui est connu aujourd’hui comme la demi-mesure de Levin.

Avant qu’on conçoive notre approche expérimentale, la demi-mesure de Levin était une distribution préalable des probabilités, qui était l’objet de vives critiques de la part des anti-bayésiens (par ex. Kevin Kelly, à la base parce qu’elle n’avait aucune justification pour être au-dessus d’une autre, déconnectée qu’elle était de tout support empirique). Notre recherche a rendu cette mesure concrète (ou si vous préférez, a fourni une version empirique de la distribution), justifiant l’idée qu’on peut produire de la structure à partir de règles aléatoires via une computation universelle (sans avoir besoin de rien d’autre).

L’idée théorique est assez simple. Pensons à la constante mathématique π . Si vous voulez produire les décimales de π en utilisant une machine à écrire, vous devez taper un chiffre après l’autre. Cependant, si vous avez un

ordinateur, vous pouvez écrire un (court) programme produisant tous les chiffres de π . Il y a donc une probabilité substantiellement plus grande de produire un programme par hasard que de produire les chiffres de π un par un. La justification expérimentale est maintenant l'approximation numérique et la simulation que nous avons entrepris en nous basant sur ces fondements théoriques. Notre recherche l'avait comme principal objectif pour fournir une méthode d'évaluation alternative afin de calculer la complexité algorithmique de chaînes courtes (un problème connu dans la théorie algorithmique de l'information), mais avec cet autre important corollaire.

5.9 Vers une conception informatique du monde

On ne peut pas nécessairement déduire de cela que le monde est algorithmique, mais je prétends que le genre de règles produisant ce type de distribution peut être exécuté par une computation digitale, et donc, selon cette hypothèse, on n'a pas besoin de convoquer d'autres propriétés pour expliquer le degré d'organisation du monde, et supposer un univers analogique ne fait que compliquer gratuitement la théorie. Pourtant cette vision algorithmique n'exclut pas d'autres possibilités.

Les visions du monde ne s'excluent pas toujours mutuellement ; certaines visions du monde en contiennent d'autres. Il n'est pas surprenant, à la lumière de ce point de vue algorithmique, que là où D'Arcy Thompson voyait des mathématiques dans les formes de la nature, Max Tegmark trouvait que notre univers était pleinement descriptible mathématiquement ; ou bien là où Zuse voyait un réseau calculateur comme illustration du monde, Fredkin a vu plus tard le monde pleinement décrit par un automate cellulaire. Pas une surprise non plus quand Wolfram explique le monde en termes de programmes simples, ni quand Lloyd comprend l'univers comme un ordinateur quantique géant. Une importante caractéristique des visions du monde est qu'elles ont aussi peu d'obligations ontologiques que possible, et si elles aspirent à être fécondes, elle doivent être supportées par une garantie empirique ou évidence expérimental, comme nous pensons l'avoir fait pour la nôtre.

Si le générateur de l'univers est algorithmique, la plupart des possibles univers seraient ordonnés et développeraient des règles pouvant s'identifier à des lois. De plus, la vision algorithmique indique que de possibles univers manifesterait concrètement une certaine distribution des structures. Si on suppose une vision computationnelle-informationnelle (digitale) de l'univers,

alors la vision algorithmique est incontournable. La vision algorithmique arrive à grouper ce genre de visions du monde, sans même exclure les modèles non-discrets ou non-numériques tant que le monde, comme l'expérience le suggère fortement, reste surtout algorithmique au sens où on l'a expliqué ici, un monde où les événements se produisent de façon intelligible. Le modèle n'a pas besoin de supposer la discrétion en tant que base pour le monde, car la vision algorithmique n'a besoin de rien d'autre que de ce type de règles exécutées par des programmes de calcul (numériques), ni d'aucune autre supposition quant à la distribution des données ou des programmes. Donc si cette vision est accompagnée de justifications expérimentales, il serait sage de conserver la théorie sous sa forme minimale, à moins d'avoir une bonne raison de la compliquer.

Le point de vue présenté ici vise à construire une explication algorithmique pour le processus grâce auquel nous voyons de la structure émerger de par le monde, cette structure qui se trouve tout autour de nous, de la formation des galaxies à l'apparition de phénomènes organisés comme la vie. On commence par se demander à quoi ressemble l'univers en terme de distribution de structures. On peut ensuite en conclure soit que le monde est numérique car il paraît l'être (ou non), soit qu'il est algorithmique (au sens numérique) même s'il n'est pas numérique, auquel cas je crois qu'il n'y a pas de raison de penser qu'il n'est pas numérique. Nous avons produit certains indices empiriques en faveur de la ressemblance entre données empiriques et numériques, et les moyens de continuer l'investigation (une investigation qui a déjà généré certaines applications, notamment le calcul de la complexité de chaînes courtes, là où les algorithmes de compression ont échoué). Parmi les idées que je défends ici, il y a l'idée que le monde aurait plus de chances d'avoir l'air aléatoire (ou d'être plus aléatoire) s'il était analogique. Si on prend des chiffres au hasard dans un monde analogique (infiniment divisible) et si ce monde hypothétique permet un véritable aléatoire (indéterministe), contrairement à un monde numérique, alors il faudrait s'attendre à ce que tous les chiffres ressemblent aux chiffres d'un Ω de Chaitin (voir la définition dans l'Appendice de cet essai). C'est un nombre qui est aléatoire par définition, sous notre modèle standard de computation. On peut réaliser la même expérience de pensée avec un alignement de nombres réels et découvrir que la chance de tirer un nombre aléatoire parmi tous les nombres réels d'un intervalle fini est de 1, ce qui veut dire qu'il y a une complète certitude de tirer un nombre aléatoire. Pourtant cela ne correspond pas à notre expérience de tous les jours. C'est complètement à l'opposé, en fait. Chaitin a prouvé qu'on ne pouvait pas calculer la plupart des chiffres d'un nombre Ω (et pour certains nombres Ω , pas même un seul chiffre), ainsi dans un monde où les nombres aléatoires persistent, les choses

peuvent avoir beaucoup de chances de ressembler à des Ω de Chaitin. Le fait qu'on puisse faire de la science dans ce monde semble être une indication qu'on n'est pas dans cette situation.

Deux outils tirés de la théorie algorithmique de l'information sont pertinents pour expliquer la présence de structures organisées dans le monde, sans nécessairement violer aucun principe physique courant et connu, et effectivement ils fournissent des explications concrètes et raisonnables sur certains de ces principes (par ex. sur la thermodynamique), en faisant l'unique supposition que ce qui arrive dans l'univers est le résultat de l'application de règles (qui, couche après couche, peuvent paraître très compliquées mais qui sont simples à l'origine, étant du type de celles qui peuvent être exécutées par des calculateurs digitaux). Si le temps dans l'univers est vu comme un temps computationnel, alors la théorie de la probabilité algorithmique, avec son concept de distribution universelle de Levin, prise avec la profondeur logique de Bennett, devrait fournir une explication de l'univers organisé dans lequel nous semblons vivre. Comme je le défends, la vision que le monde est algorithmique dans les termes décrits ici, s'appuie sur au moins deux indicateurs : l'un est la compressibilité des données de notre monde, comme prouvé par notre succès pour compresser ces données (que ce soit dans les dépôts digitaux ou analytiques). Pas seulement les données, mais aussi les lois physiques gouvernant notre réalité se trouvent être compressibles à l'aide de modèles et de formules que les scientifiques utilisent pour court-circuiter les phénomènes physiques afin de faire des prévisions sur le monde. Le second indicateur en faveur de la vision algorithmique est la distribution des structures dans notre monde quand on la compare (comme on l'a fait) à la distribution des structures produites par des mondes purement algorithmiques (en utilisant des machines abstraites).

Chapitre 6

Mécanique algorithmique des processus du vivant

La Mécanique statistique a commencé à jouer un rôle de plus en plus important pour quantifier le théorie de l'évolution comportementale. Mais tandis que l'évolution inspirait des méthodes algorithmiques d'optimisation heuristique, peu de choses ont été réalisées dans le sens de l'utilisation des concepts de la physique computationnelle pour faire avancer la compréhension des aspects saillants de l'évolution biologique et des processus vitaux. La physique des systèmes biologiques n'est pas pleinement comprise et une théorie de la computation biologique sera nécessaire pour intégrer les notions tirées de la théorie de la computation et de l'information algorithmique, afin d'éclairer des processus tels que l'émergence et la persistance des structures et des fonctions, la robustesse de systèmes hautement adaptables et évolutifs, et les caractéristiques vitales hors équilibre. Les modèles computationnels statistiques vont fournir de nouvelles pistes. Je propose, dans le cadre d'hypothèses raisonnables, qu'on puisse tracer et commencer à construire des directions intéressantes de recherche pour dévoiler les principes fondamentaux concernant le grand défi qu'est de mieux comprendre les processus du vivant. En retour, cela va contribuer à constituer des approches innovantes et alternatives dans la recherche en science du vivant et en biotechnologie.

En physique, les interactions entre particules suivent des lois. En biologie, les interactions entre biomolécules servent une fonction. Ces points de vue très différents ont commencé à fusionner quand la recherche de ces dix dernières années démontré comment, dans certains cas exemplaires, les lois de la physique contraignent les implémentations des fonctions biologiques. Hopfield s'est demandé pourquoi la biologie semblait si différente de la phy-

sique [Hopfield, 1994], plaidant pour une approche computationnelle qu’il a lui-même entreprise. Un élément central des systèmes vivants se trouve être quelque chose de digital : les séquences ADN raffinées par l’évolution encodent les composants et les processus qui guident le développement des organismes vivants. Il est donc naturel de se tourner vers l’informatique, qui possède des concepts mis au point pour caractériser l’information digitale, mais aussi vers la physique du calcul afin de comprendre les processus du vivant. Dans le sous-domaine du calcul lié à l’évolution, par exemple, l’évolution est mise en correspondance avec des algorithmes d’optimisation, et si l’évolution biologique se trouve être computationnelle par nature, on peut s’attendre à ce qu’elle soit soumise aux mêmes forces et contrainte aux mêmes lois qui gouvernent la physique et la computation. Ce que je propose, c’est que des concepts-clé de la théorie de la computation – en particulier la théorie de l’information, l’aléatoire algorithmique et la thermodynamique de la computation – puissent expliquer des aspects de l’évolution et du comportement. Je vais me concentrer sur quelques caractéristiques importantes de la vie et de l’évolution (auto-assemblage, robustesse, optimisation de l’information et de l’adaptation, compétition, ainsi que prévision et aléatoire en biologie) dont je prétend qu’elles sont des propriétés qui caractérisent aussi les systèmes computationnels. Ce que je fais consiste à utiliser des idées tirées de la physique computationnelle pour découvrir de nouvelles approches afin de comprendre et modéliser le comportement des systèmes vivants et l’évolution de la vie.

Il est important de mettre l’accent sur le fait que ce programme de recherche n’est pas seulement descriptif mais aussi normatif, au sens où une fois montré que des lois computationnelles pilotent aussi les processus des systèmes vivants, cela voudrait dire que les distributions biologiques devraient être en accord avec les mêmes principes généraux de la computation, d’où la possibilité de faire des prévisions au niveau des distributions et à des niveaux d’échelle plus élevés.

6.1 Émergence dans les systèmes biologiques auto-assemblés

Traditionnellement, en physique statistique, on dit que les transitions de phase apparaissent comme le résultat de différentes forces orientées dans des directions différentes (parfois opposées) – le volume versus la pression ou la température versus le mouvement. En tant que partie intégrante de ma recherche, il y a une enquête sur la relation entre l’approche algorithmique de

l'émergence et certains phénomènes à caractère mécanique-statistique, par ex. les phases de transitions et autres concepts moins formalisés tels que l'auto-organisation : des mécanismes que Kauffman [Kauffman, 1993] a affirmé être responsables de la complexité des organismes vivants lors de la sélection naturelle darwinienne. Mon but est de montrer que la computation joue un rôle important en tant que force qui travaille a contrario dans la sélection naturelle, c'est-à-dire une force contraignante façonnant des formes et des structures, et montrer ainsi que la plupart des choses qui se produisent même dans le monde vivant peuvent se comprendre dans des termes concernant le plus important héritage de Turing : le concept de computation universelle.

L'auto-assemblage est un aspect central de la sélection naturelle darwinienne, mais ce n'est pas seulement une caractéristique de la sélection naturelle. L'ordre visible dans la nature est largement le résultat d'auto-assemblages – des atomes de cristaux aux cellules des amas de galaxies, pour ne mentionner que quelques exemples. Parmi les plus récentes découvertes en biologie moléculaire, il y a celle sur les gènes formant des blocs de construction à partir desquels les systèmes se bâtissent [Shetty et al., 2008], une trouvaille qui apporte un nouvel éclairage sur les principes courants à la base du développement des organes en tant que composants fonctionnels, plutôt que simplement des ingrédients biochimiques. La théorie de l'évolution fonctionne comme un grand principe organisateur, mais comme cela a déjà été souligné dans le passé [Mitchell, 2011, Chaitin, 2011], il lui manque une théorie mathématique formelle générale, malgré tous les efforts respectables entrepris pour combler ce déficit [Grafen, 2007, Grafen, 2008].

Récemment, on s'est intéressé à la “mise en forme” d'un système auto-assemblé en tant que résultat d'un processus computationnel [Adleman, 1999, Rothmund et al., 2004, Adleman et al., 2001, Rothmund and Winfree, 2000, Aggarwal et al., 2005]. Ces types des processus avaient déjà été modélisés en utilisant la computation [Winfree, 1998a], et le concept d'auto-assemblage a été étudié par les informaticiens depuis von Neumann, qui travailla sur les caractéristiques des systèmes computationnels capables de fournir des structures auto-reproductrices persistantes en tant qu'un des aspect essentiel du vivant, notamment en se servant d'automates cellulaires. Finalement, ces études ont produit des systèmes montrant de nombreuses caractéristiques des processus du vivant et qui sont profondément liées au concept de computation universelle (de Turing) [Gardner, 1970, Langton, 1986, Wolfram, 2002]. Certains modèles artificiels d'auto-assemblage [Rothmund et al., 2004], par exemple, montrent tous les caractères nécessaires à la computation Turing-universelle et sont

capables de produire des formes arbitraires [Winfree, 1998b], comme si on avait affaire à un système biomoléculaire Turing-universel.

La sélection naturelle fonctionne bien pour les systèmes qui ont déjà évolué jusqu'à devenir des systèmes avec réplication héritable. Darwin savait (et l'avait admis) que sa théorie ne pouvait pas expliquer l'origine de la vie. La sélection naturelle entre en jeu uniquement après que la vie ait émergé. Dans ses expériences informatiques de 1953, Nils Barricelli avait déjà conclu que "mutation et sélection seulement ... [démonstration] sont insuffisantes pour expliquer le phénomène d'évolution" [Barricelli, 1963]. Les ingénieurs de Princeton (où Barricelli travaillait) chargés de faire tourner les expérimentations de Barricelli lui ont posé des questions, croyant qu'il y avait quelque chose dans son code dont ils n'avaient pas été informés. George Dyson [Dyson, 1998] interprète cela comme le reflet d'une surprise face au fait qu'un simple ensemble d'instructions puisse générer autant de complexité apparente. Comme Wolfram l'a montré [Wolfram, 2002], un système comme le Jeu de la Vie ne possède aucune caractéristique spéciale.

Ce passage de Dyson suggère que la biologie a été portée sur les ordinateurs dès qu'ils ont été construits, les ordinateurs ont convoqué le concept de sélection naturelle dans les machines dès le premier moment. L'idée principale de Barricelli est ce point de vue sur le monde, particulièrement sur les processus vitaux et la vie elle-même, en tant que logiciels.

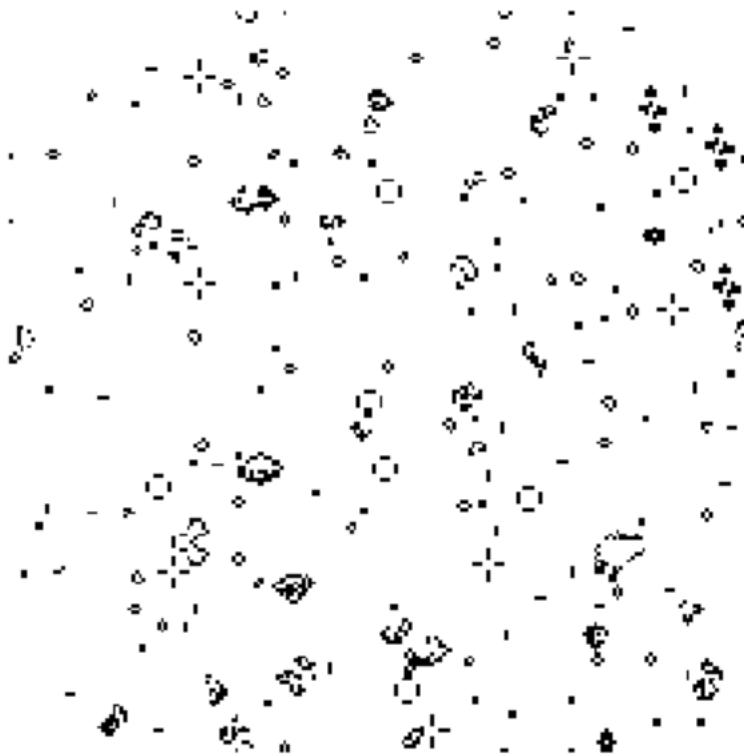


FIGURE 6.1 – La configuration d’une étape du Jeu de la Vie de Conway [Gardner, 1970] (chaque cellule regarde ses voisines pour savoir si elle reste en vie ou meurt – reste noire ou blanche). De façon assez surprenante, des processus simples comme ce système de règles, aujourd’hui appelé automate cellulaire, peuvent capturer des aspects fondamentaux de la vie comme l’auto-assemblage, la robustesse et l’auto-reproduction, et sont capables de la plus importante caractéristique de la computation : l’universalité de Turing. Von Neumann a cherché à modéliser un des plus basiques processus de l’évolution – la reproduction – en concevant ces types de règles basées sur des matrices, où l’espace est mis à jour partout en même temps par des étapes discrètes, pour étudier l’auto-reproduction.

6.2 L'émergence comme un phénomène algorithmique

Dans la vaste littérature sur le sujet, l'émergence surgit dans des contextes largement disparates, et peu de gens sont d'accord sur son exacte signification. Barry Cooper a récemment suggéré [Cooper, 2009] que le concept d'émergence pourrait bien s'expliquer en termes de computabilité théorique. J'ai une position semblable, quoique distincte. Ma proposition est que la notion d'émergence peut être saisie dans le cadre de la théorie de l'information algorithmique sans avoir recours à des modèles plus puissants de computation.

Définir l'émergence a toujours été une affaire controversée. On a souvent prétendu que les phénomènes émergents résultaient d'interactions basiques entre des agents simples, interactions produisant une complexité comportementale qu'il était difficile, sinon impossible, d'expliquer en termes d'agents dits individuels. On sait que les systèmes calculants capables d'universalité de Turing ont des propriétés qui ne sont pas prévisibles en principe. Et en pratique c'est la même chose, car des années de tentatives variées pour produire un comportement aléatoire n'ont pas permis de fournir une formule pour prédire l'évolution de certains systèmes calculants simples [Zenil et al., 2012c], malgré par exemple l'apparente simplicité de systèmes tels que l'automate cellulaire de Règle 30 [Wolfram, 2002], qui est à bien des égards la règle calculante la plus simple possible. Le comportement complexe est courant dans de nombreux systèmes simples, comme on peut le vérifier en cherchant presque n'importe où dans n'importe quel espace de règles suffisamment riches [Wolfram, 2002].

On peut faire des recherches sur les interactions entre les éléments dans une formule prise comme productrice de computations, par exemple les chiffres d'un nombre dont le déploiement tourne à l'étonnamment compliqué (irrationnel selon les anciens Grecs). C'est quelque chose qui est loin d'être trivial. Il semblerait que le concept d'agents avec leurs interactions ne suffise pas toujours à expliquer les phénomènes émergents simples du type trouvé par exemple dans les chiffres d'une constante mathématique, un objet simple comme π , sur lequel c'est toujours une question ouverte que de savoir s'il possède des propriétés comme la normalité, pour ne donner qu'un seul exemple. On pourrait soutenir que tous les digits dans π interagissent avec tous les digits précédemment calculés, ce qui pourrait donner cet aspect aléatoire. Mais pour certains objets mathématiques, tels que π , il existe certains types de formules appelées BBP [Bailey, 2000] qui ne requièrent aucun calcul

des digits antérieurs pour pouvoir rechercher et extraire n'importe quel digit du développement de π dans des bases 2^n pour n'importe quel n entier. Le i -digit de π dans n'importe quelle base 2^n peut donc se calculer sans avoir à calculer $i - 1$ digits de π .

Donc ce que suggèrent les formules BBP pour π , c'est qu'on n'est pas dans une situation où les étapes d'une computation, comme celle qui produit les digits de π , produisent un résultat complexe parce que les digits du développement de π interagissent entre eux pendant le calcul pour lui donner l'air aléatoire. Car avec ces formules BBP, on peut calculer n'importe quel digit sans avoir à en calculer aucun autre avant, au moins dans les versions de π présentées dans des bases qui sont des puissances de 2 (qui ont l'air tout aussi aléatoires que leur équivalentes en bases décimales). Ce que suggèrent les formules BBP, c'est que les interactions, au moins dans ces tentatives pour produire de la complexité comme propriété émergente d'un simple objet mathématique, sont moins importantes que la computation elle-même, malgré sa simplicité (bien sûr, π n'est qu'un exemple de nombre irrationnel calculable parmi la quantité innombrable d'entre eux dans R).

Est-ce que la complexité apparente émergente du calcul de π diffère du type d'émergence identifiée dans d'autres systèmes, où on peut clairement dire que les agents et leurs interactions produisent de la complexité d'une certaine façon ? Barry Cooper a établi que "l'essentiel situé derrière cette production surgissante était à la fois l'émergence au grand jour – à la connaissance consciente – du contenu computationnel du monde, et à la fois son ouverture à être saisissable dans des prévisions mathématiques". Je pense vraiment que les formes d'émergence auxquelles il est fait référence sont de même type, et que le principe situé derrière elles est algorithmique par nature.

Cette étude se propose d'entreprendre la caractérisation de l'idée d'émergence, au moins en jouant de deux moyens complémentaires, en faisant s'accorder des caractéristiques reconnues de l'émergence comme l'irréductibilité et la robustesse, avec ce que je suppose être une propriété saillante de certaines interactions, c'est-à-dire l'indépendance. D'un côté, la probabilité algorithmique dirige la distribution des structures, et de l'autre côté, l'aléatoire algorithmique est requalifié comme source de structures inattendues à la limite de la calculabilité. Ce projet peut se voir comme le premier pas pour essayer d'aller vers une sorte de modélisation mathématique, nécessaire pour fournir le cadre d'une discussion formelle au sujet du phénomène d'émergence, et je suggère que la discussion porte sur la nature algorithmique du phénomène, en relation avec le monde concret.

6.3 Probabilité algorithmique et robustesse biologique

Pour que la sélection naturelle fonctionne, il doit y avoir un porteur d'information capable d'accumuler les données dans le temps, pour les transférer aux nouvelles générations. Un vif point d'intérêt est l'utilisation évidente de l'ADN par toutes les formes de vie sur Terre, et certainement une nécessité qu'ont les organismes vivants de l'avoir pour stockage, en particulier comme stockage digital. L'ADN n'est pas la seule forme de stockage dans les systèmes vivants, mais c'est à ce jour la seule source commune d'information digitale héritable entre générations et populations.

L'unité de la vie terrestre est remarquable à bien des égards ; nous partageons une histoire commune en plus des éléments physiques et chimiques communs. A partir d'un simple point de départ de la sélection naturelle, on peut logiquement conclure que ce système de stockage digital a été assez “fiable” et “robuste” pour transférer l'information de générations en générations, supportant par là le processus de la vie, tout en étant en même temps suffisamment souple pour produire de nouvelles formes de vie et pour s'adapter aux changements environnementaux.

Relié à cette observation, il y a le fait important que malgré les processus aléatoires qui jouent un rôle important dans l'évolution – sous la forme de ce qui est identifié comme les mutations, qui mènent aux changements et sont finalement responsables de l'adaptation – les sources aléatoires d'erreurs n'ont pas corrompu l'information dans l'ADN. Il faut se rendre compte qu'une telle propriété est caractéristique des dispositifs de stockage digitaux mais pas des stockages analogiques, qui sont plus sujets aux erreurs. Cela pourrait expliquer pourquoi la nature a fait évoluer des formes de vie dotées d'information héritable digitales. Une question essentielle concerne le degré et le type d'aléatoire des mutations. Les simulations informatiques réalisées dans le cadre de recherches sur la vie artificielle ont reproduit différents caractères connus de l'évolution biologique [Gardner, 1970, Langton, 1986, Wolfram, 2002], tous s'avérant être profondément connectés au concept de computation universelle (de Turing). L'évolution semble manifester certaines propriétés fondamentales de la computation, et elle ne fait pas que ressembler à un processus algorithmique, elle semble aussi produire souvent le même type de distribution de résultats qu'on pourrait attendre d'une computation [Li and Vitányi, 2008].

Parmi plusieurs propriétés-clés des organismes vivants, il y a leur robustesse face aux variations génétiques et à l'évolution [Wagner, 2013]. Je

pense que les outils tirés de la probabilité algorithmique pourraient s'appliquer fructueusement à la tâche consistant à dessiner une perspective computationnelle de la vie et de l'évolution. Nous rappelons que la probabilité algorithmique [Solomonoff, 1964, Levin, 1974] peut se définir comme $m(s) = \sum_{p: M(p)=s} 2^{-|p|}$, où $m(s)$ est la somme des probabilités de tous les programmes pour lesquels M , associé à p , donne pour résultat la chaîne s et s'arrête. Comme p est lui-même une chaîne binaire, $m(s)$ est la probabilité que le résultat de M soit s quand M est alimenté en entrée avec une séquence de bits uniformément distribuée, avec un programme aléatoire et autodélimité, tournant sur une machine de Turing universelle préfixe-free.

Considérons la version biologique, une boîte avec chacun des quatre nucléotides de l'ADN : A, T, C et G. On relie la boîte à une machine, et arrive dans la machine – en tant qu'entrée – un programme aléatoire avec des instructions pour assembler ces nucléotides, afin de former une séquence ADN. La machine assemble les nucléotides sans préalable. Ce scénario ressemble beaucoup à ce qui se passe dans une cellule, avec les ribosomes compilant l'ADN en protéines, et il correspond à une structure de traitement de l'information, où les principes de la computation s'appliqueraient.

Si on désire produire les digits de la constante mathématique π en tirant des digits au hasard, il va falloir essayer pendant un bon moment avant d'obtenir quelques nombres consécutifs s'accordant avec le segment initial de l'expansion de π . La probabilité de réussir serait très faible : $1 / 10$ multiplié par le nombre désiré de digits, en base 10. Cela signifie une probabilité de seulement $(1/10)^{2400}$ pour un segment de seulement 2400 digits de π . Mais si au lieu de tirer les digits au hasard, on tire au hasard les bits d'un programme informatique, et qu'on exécute ces programmes sur un ordinateur digital, les choses s'avèrent être très différentes. Un programme produisant les digits de π a plus de chances d'être produit par un autre programme informatique tiré au hasard, que directement les digits au hasard. C'est la propriété principale de $m(s)$ – que des programmes informatiques tirés au hasard produisent des structures en filtrant une partie de ce avec quoi on les a alimentés. Partir avec une chaîne apparemment aléatoire et faire tourner sur elle un programme tiré au hasard, et il y a de bonnes chances pour que cette chaîne d'apparence aléatoire devienne une chaîne régulière et hautement structurée (comme l'est π , malgré l'apparent aléatoire de la distribution de ses digits). Les programmes produisant des chaînes structurées peuvent être dans le genre des blocs de construction utilisés dans la nature [Shetty et al., 2008], conçus pour être communs aux être vivants, en étant constitués des mêmes unités de base. Au contraire, si on tirait des particules au hasard, les chances qu'elles se regroupent de la façon dont elles le font, s'il n'y avait pas de lois physiques,

sont si minces que rien ne se passerait dans notre univers. Les lois physiques, comme les programmes informatiques, font en sorte que les choses se passent de manière ordonnée. Par exemple, dans une expérimentation avec des organismes digitaux conçus pour montrer comment des caractères complexes trouvent leur origine à partir de mutations aléatoires et de sélection naturelle, on a démontré que la plupart des chemins évolutionnaires construisent d'abord des états intermédiaires pour arriver aux mêmes fonctions complexes évoluées [Lenski et al., 2003].

La robustesse signifie que le comportement d'un système va rester qualitativement inchangé même si le système est perturbé. En biologie, cela se traduit, au plus bas niveau, par la capacité d'un organisme à préserver son phénotype en dépit des perturbations environnementales et des mutations, mais aussi, à un plus haut niveau, par l'existence de processus de développement. Dans un monde de processus calculables, $m(s)$ (1) établit que des structures simples résultant des processus simples sont probables, tandis que des structures compliquées, à l'aspect aléatoire, produites par des processus compliqués (de longs programmes) sont relativement improbables. Cela veut dire que si on alimente avec un nombre non négligeable de programmes (par ex. des programmes de réarrangement ou de retrait au hasard), les résultats de ceux qui vont marcher vont encore produire la même fréquence de structures que l'original. En fait, $m(s)$ est aussi connue sous le nom de Distribution Universelle, car elle possède la propriété remarquable de dominer toute autre demi-mesure utilisant d'autres langages de programmation (ou d'autres machines de Turing universelles) [Levin, 1974, Kirchherr et al., 1997].

Renforçant l'hypothèse que la probabilité algorithmique peut modéliser certains phénomènes biologiques, il y a le fait que certains systèmes montrent une préférence pour certaines caractéristiques, telle que d'être bihédral sous une symétrie cyclique, et la robustesse phénotypique [Johnston et al., 2011], qui semblent s'accorder avec les résultats attendus produits dans les processus computationnels et expliqués par la probabilité algorithmique. Cette robustesse peut aussi s'observer dans la dynamique d'auto-assemblage artificielle [Griffith et al., 2005], suggérant que ce n'est pas une propriété exclusive des systèmes vivants, mais que cela viendrait plutôt d'un système simple de règles faisant apparaître par évolution des structures persistantes de haute complexité.

Parmi les processus connus pour être impliqués dans la réplication et la transmission de l'ADN, il y a entre autres la translocation chromosomique (un fragment d'un chromosome se casse et vient s'attacher sur un autre chromosome), la transcriptase inverse, l'échange de fragments de codes et le crossover, la relocalisation chimique (l'appariement par liaisons hydrogène à

une séquence complémentaire) et la dénaturation de l'ADN (la séparation de morceaux de simple hélice par la cassure de liaisons hydrogènes entre les bases). Ce sont tous des processus relativement simples. Un sous-ensemble d'opérations purement digitales peut faire correspondre ces opérations à des équivalentes computationnelles. Des opérations comme la jointure, la copie, le découpage, la complémentation, l'appariement, ou le remplacement sont équivalentes à celles observées dans l'ADN. Les règles déterminant la façon dont l'ADN se réplique, peuvent aussi être du même type algorithmique que celles qui gouvernent d'autres types de phénomènes physiques. La construction de l'ADN semble venir d'une longue période d'applications des règles simples, couches après couches, le code du vivant s'est construit sur des milliards d'années au cours d'un processus profondément algorithmique possédant ses propres règles caractéristiques, faisant en sorte que des choses comme le déploiement des protéines nous paraissent complexe. Cependant, la structure des protéines peut montrer un haut degré de régularité [Li et al., 1996], formant un nombre limité de symétries plutôt que des configurations aléatoires. Plus important encore, ces régularités sont liées à la stabilité évolutive et semblent distribuer les structures conformément à la probabilité algorithmique $m(s)$ (voir page 668 dans [Li et al., 1996]).

6.4 Modèles computationnels de taux d'évolution biologique

Le type de “robustesse algorithmique” sur lequel faire des recherches peut correspondre, par exemple, à des modèles où la partie mutation aléatoire est moins importante que les mutations intervenant au niveau des instructions qui déterminent les fonctions biologiques, pouvant par là finalement expliquer des processus de spéciation favorisant des phénomènes tels que l'épistasie et les modèles d'équilibres ponctués, et aussi expliquer la convergence dans l'évolution (des changements évolutifs dans des lignées distantes les unes des autres mais qui donnent des formes similaires). Car si l'adaptation par sélection naturelle a le pouvoir de produire toute une variété de traits, pourquoi devrait-elle produire ce genre de répétitions face à des paramètres environnementaux semblables ? Évidemment, il semble que cela ait quelque chose à voir avec un processus autre que juste une exploration des combinaisons, et cela vient probablement en complément des pics adaptatifs forcés par des contraintes physiques. La réponse algorithmique est qu'il y a beaucoup moins de programmes possibles et de variations de programmes possibles (changements de sous-routines), que de solutions combinatoires possibles.

Une question associée est de se demander s’il n’existe rien dans l’évolution comportementale des systèmes simples calculants, qui puisse modéliser les taux de changements subis par les traits d’un organisme. C’est la question de la vitesse de l’évolution, les occasions où cela semble fonctionner plus vite (par ex. l’explosion cambrienne), et les moments de remarquables stases pour de nombreuses espèces sur de longues périodes, malgré une évolution phénotypique rapide et de la différenciation génétique (voir par ex. [Catania, 2010]). A cet égard, nous avons trouvé qu’il existe une approche computationnelle pour étudier les taux possibles de changements sur des échelles de temps évolutives, comme dans les stases et les évolutions phénotypiques rapides [Bennett, 1997], une approche pouvant être explorée en termes de phénomènes ralentis ou accélérés dans de petits programmes informatiques mis en compétition pour calculer une fonction [Zenil, 2010]. Les accélérations ressemblent parfois beaucoup à des explorations combinatoires dans un espace logiciel, et d’autres fois, elles font penser à ces petits changements dans un programme capables de produire d’importants changements dans l’évolution d’une espèce tout en préservant sa structure. Notre étude a révélé des structurations variées dans le microcosme des petites machines de Turing. On a trouvé qu’en moyenne, les fonctions tendent à être calculées plus lentement quand plus de ressources est ajouté. Plus remarquablement, on a observé des “transitions de phase” dans la distribution des probabilités d’arrêt (des sautes de temps que les machines prennent lorsque on ajoute de ressources), en accord peut-être avec des découvertes récentes de transitions adaptatives accélérées dans l’évolution des multicellulaires [Ratcliff et al., 2012] ou on a parfois des temps d’évolution accélérée.

6.4.1 Le coût du traitement de l’information et de l’aptitude

De nombreux défis fondamentaux en écologie requièrent de comprendre comment un changement environnemental affecte la dynamique de populations interactives. Si on peut regarder les systèmes biologiques comme des systèmes computationnels, on pourrait chercher à l’intérieur des systèmes biologiques comme on le ferait dans des systèmes computationnels. Si les concepts algorithmiques peuvent s’adapter avec succès au contexte biologique, les concepts-clé développés en théorie de la computation pourraient non seulement être pertinents en biologie, mais ils pourraient aussi expliquer des aspects fondamentaux de processus biologiques.

Inspirés par le récent travail de McNamara et

Dall [McNamara and Dall, 2010], on a entrepris une étude sur les compromis entre les bénéfices issus d’une acquisition d’information et l’énergie dépensée pour la traiter [Landauer, 1961], en trouvant et quantifiant les liens, les taux d’échanges et les valeurs de dissipation, par l’utilisation de la théorie de la thermodynamique computationnelle (la relation entre computation, information et énergie).

Pour saisir le rôle de l’information dans les systèmes biologiques, pensons à un ordinateur comme système idéalisé de traitement d’information. Convertir l’énergie en information est aujourd’hui assez facile à comprendre d’un point de vue pratique [Landauer, 1961, Feynman et al., 2000, Bennett, 1982]. Les ordinateurs peuvent se concevoir comme les moteurs transformant l’énergie en information. Il suffit de connecter un ordinateur au réseau électrique pour qu’il réalise une tâche et produise une nouvelle information.

Il est clair que les animaux convertissent l’information en énergie – par exemple, en se servant d’information pour localiser de la nourriture [Paz Flanagan et al., 2011], ou pour apprendre à chasser [Catania, 2010]. De même, il est clair que l’information peut servir à extraire des quantités encore plus grandes d’énergie à partir de l’environnement – en investissant dans la construction et les opérations d’un cerveau. Il découle de cela que la façon dont un organisme s’adapte ou est capable de se reproduire dans un environnement particulier, dépend d’un ratio positif d’échanges entre information et énergie.

Cela fonctionne aussi dans l’autre sens, dans la conversion d’information en énergie, qui a été étudiée en thermodynamique de la computation. Un bon moyen de comprendre ce processus est d’utiliser la fameuse expérience intellectuelle connue sous le nom de Paradoxe du Démon de Maxwell [Bennett, 1982, Calude and Meyerstein, 1999]. L’idée de base est qu’on peut se servir de la connaissance des micro-états d’un système pour le rendre plus chaud. Se servir de bits pour produire de l’énergie est bien expliqué dans [Landauer, 1961] et [Sethna, 2006]. Les relations avec la théorie de l’information et avec la complexité algorithmique, sont cependant moins bien comprises, mais on a le sentiment de bien savoir comment connecter ces concepts grâce à la théorie de l’information algorithmique.

Quand on échange de l’information pour de l’énergie, et de l’énergie pour de l’information, une certaine énergie s’échappe inévitablement dans l’environnement. Donc c’est aussi un sujet concernant l’écologie et finalement le changement climatique. La dissipation est un phénomène général dans le monde réel, et il nous dit que quelque chose est perdu durant le processus d’échange, quelque chose qui interagit avec l’environnement affectant

d'autres organismes. En accord avec le second principe de la thermodynamique, on peut voir cela comme de l'information sur l'irréversibilité du système. En biologie, ce type d'échange se produit sans arrêt. Les neurones, par exemple, dissipent environ $1011kT$ par décharge [Bennett, 1982]. Les ordinateurs (surtout à cause de leur dispositif de mémoire volatile – la mémoire RAM) dissipent aussi de l'énergie correspondant à au moins $1kT$ par bit [Landauer, 1961, Landauer, 1991, Bennett, 1982] (avec k la constante de Boltzmann et T la température ambiante), raison pour laquelle les ordinateurs chauffent et ont besoin d'un ventilateur interne.

Dans un article récent [McNamara and Dall, 2010], McNamara et Dall ont montré que, en faisant certaines suppositions, le traitement de l'information par des individus pourrait n'être qu'une propriété d'accroissement de l'aptitude. Il s'en suit que tout système ayant accès à une quantité finie de ressources (par ex. la mémoire) subit un coût, sans impliquer autre chose que du traitement d'information. Donc, si un organisme a pour but de récolter de l'information sur le monde, en mettant à jour ses états antérieurs par le remplacement de l'information ancienne par de la nouvelle (apprentissage), alors un coût est inévitable, et prend la forme d'une dissipation. La relation avec l'aptitude devient alors naturelle. Les systèmes vivants peuvent se voir comme des systèmes ayant accès à des ressources finies, et sujets aux mêmes coûts que n'importe quel autre système computationnel et physique.

Comme cela est souligné en [de Vladar and Barton, 2011], la sélection naturelle peut être considérée comme une extraction d'information à partir de l'environnement pour la coder dans une séquence ADN. Mais les efforts courants en vue de quantifier le contenu informatif ne s'aventurent habituellement pas au-delà de la théorie de la communication de Shannon [Shannon, 1948]. Néanmoins, comme je le suggère, des analogies avec la computation et la complexité algorithmique peuvent nous aider à comprendre comment la sélection accumule et échange de l'information et de l'énergie vis-à-vis de l'environnement lors du remplacement des populations, comme ces analogies nous avaient déjà aidés à relier la computation et la physique en thermodynamique. Pour répondre à ces questions liées à l'information, il n'y a pas de meilleur choix que d'utiliser la théorie classique de la computation et la théorie de l'information algorithmique, dans leur versions actuelles.

6.5 Recherche et formation de structures biologiques

La computation évolutionnaire, c'est avant tout de la recherche. En informatique, les algorithmes de recherche définissent un problème en termes de recherche, où l'espace de recherche est un espace rempli par toutes les solutions possibles du problème, et un point dans cet espace définit une solution. Il existe plusieurs types d'algorithmes de recherche, parmi lesquels les algorithmes évolutionnaires de recherche constituent un sous-ensemble important et fécond.

Comme l'a montré une expérience sur *eschericia coli* [Barrick et al., 2009], l'évolution survient même dans des environnements constants, ce qui suggère que l'évolution n'est pas seulement conduite par l'adaptation ou l'environnement, ni par la compétition quand les ressources sont à disposition, laissant entendre par là que l'évolution pourrait bien être une recherche au hasard.

On est toujours persuadé que le niveau de complexité observé en biologie doit, d'une certaine manière, être uniquement associé à la sélection naturelle. Mais la recherche sur la vie artificielle a donné l'occasion de faire plusieurs simulations informatiques détaillées de sélection naturelle, et ces simulations ont reproduit diverses caractéristiques connues de l'évolution biologique. Stephen Wolfram a fini par être convaincu que la plupart des complexités biologiques ont leurs racines dans des phénomènes semblables à ceux observés dans des systèmes computationnels comme les automates cellulaires – et n'ont probablement pas grand chose ou même rien à voir avec la sélection naturelle, dans le cas des structures de textures ou de pigmentation.

Les automates cellulaires sont un exemple que Wolfram a largement étudiés [Wolfram, 2002]. Chaque cellule d'un automate cellulaire peut représenter de nombreuses cellules biologiques. Le concept d'évolution darwinienne peut nous pousser à supposer que tout processus donnant naissance aux formes observées dans les systèmes biologiques doit obligatoirement avoir été soigneusement façonné par la sélection naturelle, mais le point essentiel de ces systèmes, c'est qu'il y a une règle qui pilote le comportement de ces cellules, et cette règle peut correspondre à un programme simple. Comme Wolfram le souligne, quand on regarde les différents programmes simples – dans un panorama de toutes leurs possibilités – on voit une grande diversité de comportements. Et remarquable observation, cette diversité ressemble au type de diversité observé dans les systèmes biologiques. Dans un cas comme les motifs présents sur les coquillages de mollusques, Wolfram a montré qu'il y avait une correspondance précise entre les comportements de programmes d'automates

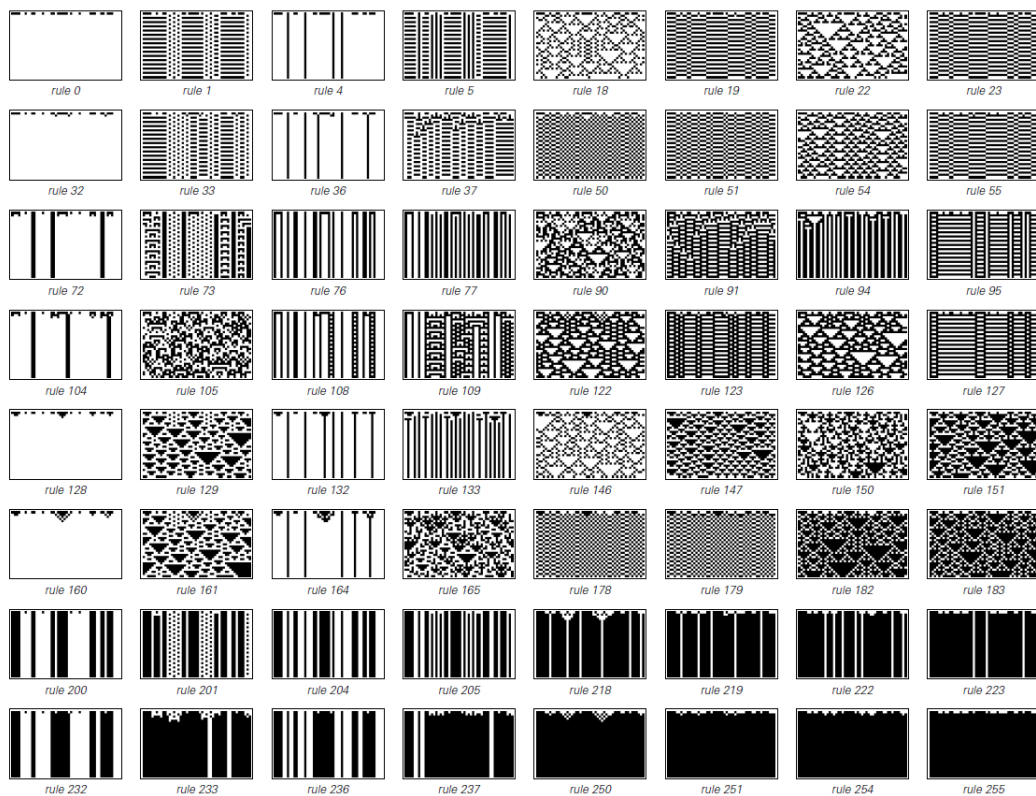


FIGURE 6.2 – Exemples de structures produites par l'évolution de quelques-unes parmi les plus simples règles possibles d'automates cellulaires, symétriques et unidimensionnels, en partant d'une condition initiale aléatoire.

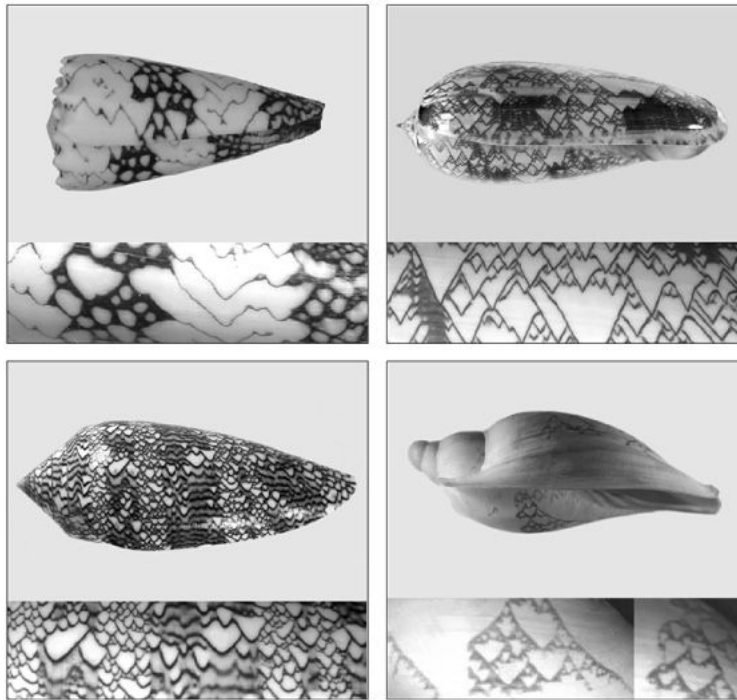


FIGURE 6.3 – Exemples typiques de motifs de pigmentation sur des coquilles de mollusques [Wolfram, 2002]. Les motifs des coquillages montrent une correspondance détaillée avec le comportement d’automates cellulaires.

cellulaires et la diversité retrouvée dans les motifs des coquillages.

Le point de vue de Wolfram, c’est que ces motifs sont, dans leur essence, le reflet de changements complètement aléatoires dans des programmes génétiques sous-jacents, sans effets systématiques venant d’une sélection naturelle. Par exemple, il est difficile de justifier les différents motifs de pigmentation des vaches en terme de sélection naturelle. Alors qu’il est facile de voir d’où cela peut venir, si la nature a récupéré, d’une façon ou d’une autre, un programme aléatoire produisant ce genre de pigmentations différentes de la peau. Ce que Wolfram a trouvé s’accorde avec mon argument disant que la complexité survient aisément en prélevant au hasard des programmes simples, et que ces programmes favorisent la reproduction de motifs robustes. Les caractères les plus évidents de la complexité dans les organismes biologiques apparaissent du fait que la nature explore l’espace des programmes de calcul qu’elle a sous la main. L’utilisation de ce type de recherche exploratoire est nécessaire à la réussite de l’évolution, mais ce n’est pas l’explication de

chacune des caractéristiques des systèmes biologiques, cette explication n'est rien de plus que ceci : les systèmes biologiques lancent de règles simples de calcul pour la simple raison que cela produit des structures qui ne leur coûtent pratiquement rien. Et c'est ça qui donne de la complexité dans les systèmes biologiques et non-biologiques.

6.6 L'universalité de Turing et le rôle de la complexité décrite dans les systèmes vivants

Le modèle Turing de computation n'était pas le premier, mais c'était le plus proche de l'esprit mécaniste du paradigme newtonien en science. La machine de Turing offrait un modèle de calculabilité des fonctions sur la gamme des nombres entiers, et fournissait un cadre pour les processus algorithmiques, avec une correspondance facile avec les structures physiques.

Parmi les nombreuses contributions faites par Alan Turing basées sur son concept d'universalité computationnelle, il y a eu l'union entre données et programmes. On pourrait penser que l'entrée d'une machine de Turing est une donnée, et que le tableau de règles de la machine est un programme, et que ce sont deux entités séparées, mais ce n'est pas le cas car ils sont interchangeables. Turing a montré qu'identifier quelque chose comme une donnée ou un programme, était simple affaire de convention, et pas une distinction fondamentale. Il est important de garder cela à l'esprit quand on considère l'univers d'un point de vue computationnel, car on pourrait assez raisonnablement se demander quelle partie représente les données et quelle autre les programmes. La réponse classique est que les programmes sont comme des lois physiques, tel que je l'ai suggéré au début. Pourtant, on n'a pas besoin de faire ce genre de distinction si, comme pour les machines de Turing, données et processus sont le même type d'objet par nature, comme le laisse entendre une vision algorithmique du monde.

Turing a fait une autre grande contribution en montrant que l'universalité arrive avec un coût, car il devient impossible de dire si, oui ou non, une computation donnée va s'arrêter, sans réaliser concrètement cette computation (en supposant qu'on ait le temps suffisant pour arriver au bout). Si la computation est l'une de celles qui ne s'arrête jamais, il n'existe aucun moyen pour que ce soit prouvé. Turing a établi que pour les systèmes computationnels, en général, il n'est pas possible de dire s'ils vont s'arrêter ou non, identifiant par là une forme d'imprédictibilité comme faisant partie de leur

nature fondamentale, en dépit du fait que ce sont des systèmes déterministes et répétables.

Comme Cooper le souligne [Cooper, 2009], on sait bien qu’en mathématiques, des descriptions compliquées peuvent nous emmener au-delà de ce qui est calculable. L’idée sous-jacente n’est pas nouvelle et vient de Hume, Leibniz et Hobbes, pour citer trois sources. Le principe sur lequel sont basés tous ces efforts est qu’on ne peut réellement comprendre quelque chose que si on est capable d’écrire un programme informatique. En écrivant un programme, il faut tout d’abord trouver un moyen précis de décrire le problème, pour ensuite concevoir un algorithme supportant le calcul qui fournira la solution. Pour ensuite atteindre cette solution, il faut lancer une machine avec le programme qui a été écrit. Mes collègues et moi avons tenté plusieurs fois dans le passé de traiter de questions épistémologiques en se servant de la théorie de l’information et d’expériences informatiques, avec certains résultats intéressants [Zenil, 2010].

6.6.1 Temps d’arrêt, et émergence épistémique contre émergence ontologique

Ronald, Sipper et Capcarrère ont conçu un “Test de Convergence” [Ronald et al., 1999] qui identifie ce qu’on attend d’un phénomène émergent. Comme noté en [Cooper, 2009], cela suit l’exemple du Test de Turing pour détecter une machinerie intelligente en étant aussi un test dépendant de l’observateur, et chose importante, il fait entrer la surprise dans la liste des phénomènes indiquant l’absence de lien causal évident, la surprise que l’observateur éprouve quand il rencontre une caractéristique du système en question. Or comme Humphreys le dit [Humphreys, 1996, Humphreys, 1997], en utilisant la notion de niveaux, il y a une hiérarchie des niveaux de propriétés $L_0; L_1; \dots; L_n; \dots$ pour lesquels au moins un niveau distinct est associé avec la matière traitée de chaque science en particulier, et L_j ne peut se réduire à L_i , pour tout $i < j$.

Comme suggéré aussi par Cooper [Cooper, 2009], on peut identifier l’ensemble des arrêts de la machine Universelle de Turing à un phénomène émergent, car ils sont incalculables, ce qui qualifie en lui-même cet ensemble en lui attribuant un effet global suffisamment surprenant. Je suis d’accord avec Cooper, et je développe ce point en me servant d’un résultat obtenu par Cris Calude et Michael Stay [Calude and Stay, 2008]. Je pense que Calude et Stay ont fourni une base pour ce qui peut se caractériser comme une surprise algorithmique, conduisant à la formalisation de l’expérience d’un

phénomène émergent, en tant que phénomène algorithmique profondément lié à l'aléatoire algorithmique. En [Calude and Stay, 2008], on observe que pour n'importe quelle distribution calculable de probabilités, les temps longs sont effectivement rares, de sorte qu'à la limite, ils ont tous le même comportement quel que soit le choix de la distribution. Ils ont prouvé que le temps exact après lequel le programme s'arrête dépende de la taille du programme et que la plupart des programmes, ou s'arrêtent "rapidement", ou ne s'arrêtent jamais.

Calude et Stay ont cependant montré que les machines de Turing "à longue durée" ne pouvaient s'arrêter qu'après des durées non-aléatoires ; la densité des durées non-aléatoires proches de n est environ $1/n$. "À longue durée" signifie que si on a une machine universelle U et que la machine M est implémentée par un programme m pour U et de longueur n , alors $U(m)$ dure plus de $c \times 2^n$ étapes, où c est une constante non-calculable dépendant de U ? d'où l'élément de "surprise". Néanmoins ces durées d'arrêt non-aléatoires sont rares (en pratique), car la densité des durées pour lesquelles un programme à N -bits peut s'arrêter décroît rapidement. Cela peut vouloir dire que la durée obéit à une distribution possédant une structure algorithmique définie, signifiant par là que les motifs ne sont pas la seule chose qui viennent d'une probabilité algorithmique, mais que les temps après lesquels ces motifs apparaissent possèdent aussi une structure et suivent aussi une distribution. Cela nous amène à une question intéressante au sujet de l'émergence ? peut-on spécifier si elle est épistémologique ou ontologique par nature, en se servant du critère calculabilité opposé à non-calculabilité. Pour ma part, je pense qu'on n'a pas besoin de dépasser les limites de la calculabilité de Turing pour saisir le phénomène. Ainsi la non-calculabilité garantit que le phénomène est irréductible par essence. En d'autres mots, la calculabilité de Turing représente la structure, tandis que l'indécidabilité du problème de l'arrêt représente la plus importante propriété de l'émergence, sa nature finalement irréductible.

6.7 Rôle et nature de l'aléatoire dans les processus biologiques

Les problèmes biologiques impliquant de l'aléatoire sont d'un intérêt particulier. l'aléatoire joue un rôle fondamental dans différents processus des systèmes biologiques. d'une part, sous forme de mutations, il permet l'adaptabilité lors de l'évolution naturelle. Il est évident aussi lors de la dérive génique, le changement de fréquence du variant d'un gène (allèle) à l'intérieur d'une population, à cause du prélèvement au hasard. Les allèles d'une

progéniture sont un échantillonnage de ceux des parents, et le hasard joue un rôle en déterminant si un individu donné va survivre et se reproduire. Il est manifeste aussi dans les environnements variables, et dans les opérations de mutations en tant que source de variations génétiques. En computation évolutive, on utilise habituellement un décalage de bit, mais les mutations naturelles peuvent impliquer de larges séquences d'ADN, qui se dupliquent (habituellement) par recombinaison génétique par exemple.

d'autre part, le hasard joue aussi un rôle dans le comportement animal, car les animaux se déplacent de façon apparemment aléatoire quand ils essaient d'éviter leurs prédateurs. Et je pense que l'aléatoire en question ici est généré dans bien des cas par des règles assez simples ?de sorte qu'en principe il devrait être prévisible. Mais on peut remarquer que l'évolution biologique n'a apparemment jamais produit de prédateurs qui soient capables d'attraper leur proie en faisant des prédictions sur quelque chose qui nous apparaît comme particulièrement aléatoire ; les stratégies tendent au contraire à se baser sur des astuces qui ne nécessitent pas de prévoir autre chose de plus que des répétitions (par exemple [Catania, 2010]).

Des altérations aléatoires sur un projet de création peuvent ne pas donner que des changements inoffensifs, mais aussi amener des modifications plus dangereuses. Toutefois, si les changements sont des mutations dans un programme déterminant la dynamique d'une évolution, le résultat va avoir un impact plus important, mais il y a aussi de bonnes chances de produire quelque chose possédant encore une structure. En d'autres termes, comme on l'a vu ci-dessus, les changements sur un programme ont peu de chances de détruire la structure car les programmes aléatoires sont suffisamment robustes pour résister à de telles variations (ils obéissent à la distribution des probabilités algorithmiques, où la plupart des programmes produisent des structures d'une complexité organisée).

En tant que conséquence du comportement aléatoire, les propriétés du vivant paraissent irréductibles et imprévisibles, comme Wolfram [Wolfram, 2002] l'a souligné. On peut caractériser ces propriétés en terme d'incompressibilité, moins un comportement est compressible, plus il est aléatoire (par ex. les vols de Lévy d'un motif aérien de déplacement). Une des principales possibilités consiste à prédire les structures émergentes en se basant sur des ensembles de valeurs de paramètres généraux. Certains travaux ont été entrepris dans cette voie [Israeli and Goldenfeld, 2006], et j'y ai personnellement contribué avec une mesure basée sur la compressibilité, afin de capturer la sensibilité d'un système face à des stimuli externes [Zenil, 2010] et afin de réaliser certaines expériences d'épistémologie pratique sur des systèmes computationnels déterministes [Zenil et al., 2012c].

6.7.1 Compétition algorithmique et prévision

En partant du point de vue que le comportement résulte d'un calcul, au sens décrit en [Livnat and Pippenger, 2006], je vais étudier les modèles prédateur-proie d'après une perspective algorithmique. Je vais proposer des modèles algorithmiques prédateur-proie dans lesquels des séquences comportementales de stratégies de fuite des proies doivent obliger le prédateur à atteindre le même niveau de sophistication pour que sa stratégie de chasse réussisse, en expliquant ce comportement et en quantifiant les avantages évolutifs d'un choix en faveur de stratégies apparemment aléatoires.

Pour s'attaquer à cette question, la capacité d'un animal à prédire le comportement d'un autre animal et à s'adapter aux changements dans son environnement va aussi être reliée à une variante importante de la complexité algorithmique, et rattachée aux données expérimentales rapportées dans [Livnat and Pippenger, 2006] quant aux connexions avec les prises de décision en tant que sources potentielles de comportement aléatoire. La capacité de prévision d'un animal est liée à la complexité de l'environnement. Par exemple, si le comportement d'un animal est suffisamment aléatoire, sans structures apparentes, un prédateur pourra le trouver difficile à suivre. En fait, le comportement du prédateur va devoir s'ajuster par rapport au temps comportemental de sa proie. On va établir des relations entre computation, information algorithmique et comportement animal, et on va étudier les compromis entre leurs différentes mesures de complexité.

Les stratégies de fuite choisies par les proies sont aléatoires (par ex. les stratégies des gazelles, antilopes, etc.), tandis que le prédateur doit se débrouiller avec ce genre de complexité (C) en essayant de deviner la séquence des déplacements de la proie. L'approche algorithmique fournit un cadre de travail pour mieux comprendre et modéliser les connexions informationnelles et le mécanisme computationnel des modèles prédateur-proie.

Ce projet va traiter comme exemples la recherche de nourriture et la chasse, car elles nécessitent souvent à la fois de prévoir le cheminement d'une proie potentielle ou de suivre son comportement aussi précisément que possible, et à la fois de prévenir toute stratégie de fuite que la proie pourrait élaborer. Certains serpents d'eau vont plus loin, comme par exemple le serpent à tentacules, qui pousse sa proie à se comporter d'une certaine façon avant de l'attraper [Catania, 2010]. Cela laisse supposer que certaines attaques sont basées sur l'anticipation des mouvements d'une proie potentielle, autant que sur le suivi des mouvements. Il est clair que dans l'un ou l'autre cas, pour réussir, la stratégie de chasse d'un prédateur doit au moins égaler en sophistication le comportement de la proie.

Par exemple, les seules façons pour les grenouilles de prédire le comportement d'un insecte volant comme une mouche, est soit de lancer l'attaque quand la mouche ne bouge pas, soit de frapper assez vite à côté de la position où est la mouche à un moment donné. Mais cette stratégie n'a aucune chance de réussir si ce n'est pas exécuté assez vite. Car comme le mouvement de la mouche est aléatoire, un prédateur, s'il y arrive, ne peut prédire son déplacement qu'une fraction de seconde à l'avance. Le couple prédateur -proie peut se voir comme s'accordant l'un l'autre au niveau complexité comportementale. Comme cela a été établi aussi en probabilité algorithmique (1), qui est liée à la complexité algorithmique, la prévision est une mesure de complexité finalement intimement associée à la complexité algorithmique.

Cette approche pourrait par exemple fournir une base objective aux fameux résultats de Reznikova [Reznikova et al., 2012] sur l'émergence de structures comportementales chez les fourmis, et nous avons déjà conçu un système pour procurer aux chercheurs les moyens d'évaluer la complexité des chaînes binaires, même courtes, lorsqu'elle apparaissent fréquemment dans des applications concrètes (comme celle de Reznikova). Un système présenté comme une méthode qui vient compléter l'approche traditionnelle des algorithmes de compression, pour faire des approximations sur la complexité d'objets discrets.

J'ai déjà entrepris des recherches pour quantifier le comportement d'un système. Premièrement en définissant son coefficient de transition pour essayer de quantifier le comportement de systèmes à l'aide de paramètres d'ordre [Zenil, 2010], et deuxièmement en fournissant une méthode pour évaluer objectivement la complexité de chaînes courtes [Delahaye and Zenil, 2012], des chaînes d'une longueur et d'un type dont on se sert dans les expérimentations comportementales (voir [Reznikova et al., 2012, Gauvrit et al., 2011]).

6.8 L'évolution biologique comme un processus algorithmique

Selon Chaitin [Chaitin, 2011], il devrait être possible de formuler une définition simple mais générale de la "vie", et de prouver que, sous certaines conditions, on peut s'attendre à ce que "ça" évolue. C'est aussi l'orientation donnée par le point de vue algorithmique de l'évolution largement préconisé par Dennett [Dennett and Mittwoch, 1996] et par Wolfram dans un contexte plus large [Wolfram, 2002].

Une des approches est d'essayer d'expliquer les fondements de la vie selon une perspective théorique algorithmique, en terme d'évolution logicielle. Par exemple, on peut définir une mutation logicielle comme l'ajout d'un bit unique, en copiant une sous-routine ou en la modifiant.

L'idée est de prendre l'espace logiciel (l'espace des programmes informatiques) et de piocher au hasard une machine de Turing universelle U comme langage de programmation généraliste, sous laquelle on va lancer des programmes informatiques. Comme U est universelle, l'espace logiciel inclut tous les algorithmes possibles pour calculer n'importe quelle fonction.

Le moyen le plus direct pour définir une métrique dans un espace logiciel, c'est le nombre de points de mutations nécessaires pour passer d'un organisme à un autre. Les mutations peuvent être de différentes formes : changer un bit, détruire un bit, ou insérer un bit. Ou alors de façon plus énergique, changer les sous-routines d'un programme, en les traitant comme des blocs de construction.

Il y a des questions pour lesquelles raisonner sur l'ADN en termes de computation [Li, 1999] peut s'avérer utile pour comprendre la vie, et sur lesquelles ce projet a l'intention d'aller plus avant :

1. Quels sont les mécanismes computationnels/informationnels qui peuvent bien soutenir les systèmes biologiques ? Et quelles notions de computation pour quels modèles d'évolution ? Par exemple, les modèles de vitesse d'évolution varient dans leurs prédictions des formes biologiques intermédiaires. D'un point de vue algorithmique, un petit changement dans un programme peut donner lieu à un énorme changement dans son résultat. Le changement dans le résultat n'a plus besoin d'être continu, il peut être extrêmement discontinu.
2. Quel est le degré de soudaineté des étapes majeures d'une évolution possible, par exemple le passage des organismes unicellulaires aux multicellulaires ? Ce qui se passe ici, c'est que le programme principal (la cellule isolée) devient une sous-routine dans un organisme multicellulaire plus grand. Transformer le programme principal en une sous-routine appelée plusieurs fois n'est pas une grosse modification logicielle.
3. Est-ce que la possibilité pour une structure d'être récursive et hiérarchique, dans les organismes vivants, est une conséquence de processus algorithmiques de basse complexité descriptive ? Comme l'a souligné Chaitin [Chaitin, 2011], la structuration en sous-routines est un bon moyen de construire une grande portion de logiciel à partir

de petits bouts de logiciels, en accord avec les modèles courants de développement [McGinnis et al., 1983, Scott and Weiner, 1984] et de croissance [Young and Corey, 1990] biologique.

4. Quels sont les coûts du traitement de l'information liés à l'aptitude, de façon à minimiser la dépense d'énergie ? Et avec quelle efficacité un organisme peut-il échanger de l'information pour de l'énergie, et de l'énergie pour de l'information, en terme de thermodynamique computationnelle ? La dissipation dans l'environnement, à partir des processus d'échanges de ressources, peut-elle être quantifiée et finalement minimisée ?
5. Quelle est la correspondance détaillée entre les paramètres mesurés et le type de définition de la complexité qu'on s'est permis d'utiliser ? Que donneraient les variantes de la mesure de complexité computationnelle ?
6. Si la nature calcule, comment fait l'évolution biologique pour choisir quels programmes utiliser ? Pourquoi certaines formes et structures semblent avoir la préférence et apparaître plus souvent dans la nature ? Ce phénomène peut-il s'expliquer par la probabilité algorithmique ?
7. Quelle est la quantité d'aléatoire mathématique présente dans la nature et quelle est son importance quant à la construction de la complexité des systèmes vivants ? L'aléatoire algorithmique peut-il quantifier l'aléatoire biologique, quel qu'en soit le degré, de l'aléatoire des mutations jusqu' à la dérive génique ?
8. Existe-t-il des mécanismes universels conduisant à la complexité ou la simplicité biologique, ou bien la complexité/simplicité survient-elle pour des raisons qualitativement différentes indépendantes de l'environnement ?

Dans la relativement courte histoire de la biologie théorique, parmi les contributions les plus innovantes, il y a celle de Turing en relation avec la formation des structures. Pour la première fois, il a fourni un modèle mathématique mécaniste pour les caractéristiques de la morphogénèse. Si Turing avait eu connaissance de l'ADN en tant que mode de stockage digital dans les systèmes biologiques, contenant les instructions pour la vie, nul doute qu'il aurait fait la relation avec la computation universelle.

Le concept de sélection naturelle peut nous laisser penser que tout processus donnant concrètement naissance aux formes observées dans les systèmes biologiques doit avoir été soigneusement élaboré par l'évolution darwinienne, mais certaines de ces formes sont parfois difficiles à justifier simplement en

termes de dynamique évolutive. De plus, l'évolution elle-même semble manifester certaines propriétés fondamentales de la computation, et elle n'a pas seulement l'air de ressembler à un processus algorithmique, mais elle semble aussi souvent produire le type de résultat qu'on pourrait s'attendre à voir produit par une computation, en l'occurrence la forme sous laquelle les plus importantes instructions concernant la vie sont stockées – sous forme digitale (l'ADN), même si c'est de façon tarabiscotée, et sous forme d'unités identifiables d'informations (les gènes), même d'il y a tous ces cheminements et connexions intriqués et compliqués. D'où la pertinence de la théorie de la computation et de la théorie de l'information.

Après la découverte de l'ADN, il devint évident qu'au moins une partie importante des systèmes biologiques et de l'évolution était de nature digitale et computationnelle, impliquant du code et des appareillages pour copier et réutiliser ce code, et pour produire mécaniquement de nouveaux composants à partir d'instructions spécifiques. En plus de la computation évolutive, et en reconnaissance pour ce que la biologie a légué à la computation sous forme de modèles inspirés par la nature, on doit admettre qu'un échange entre ces disciplines (la théorie de la computation et la biologie de l'évolution) est possible, non seulement à cause de l'existence de fortes similarités, mais aussi parce que la computation peut expliquer des aspects fondamentaux de la vie, via l'explication possible d'importants aspects de processus biologiques. On a donc raison d'espérer que cette approche consistant à explorer des problèmes biologiques en se servant de la théorie de la computation et de l'information algorithmique se révèle féconde.

6.9 Notes du chapitre

$m(s)$ est dite une demi-mesure parce que la somme des probabilités n'arrive pas jusqu'à 1 (étant donné l'indécidabilité du problème de l'arrêt) et, en tant que fonction non-calculable, elle est uniquement approchable d'un seul côté, par le bas. La demi-mesure de Levin est aussi appelée Distribution Universelle (car Levin a prouvé qu'elle dominait toute autre distribution demi-calculable, et elle a été identifiée comme la distribution universelle miraculeuse dans [\[Kirchherr et al., 1997\]](#)).

On pourrait croire qu'en changeant le formalisme, on pourrait obtenir une toute autre distribution, mais ce n'est pas le cas, comme nous l'avons aussi montré. Nous avons montré que des formalismes computationnels raisonnables donnent des mesures de complexité raisonnables. Cela signifie que

la distribution n'est pas seulement empirique (et donc n'est plus une distribution a priori) mais aussi assez saine et stable pour servir à faire des comparaisons statistiques avec la distribution des structures dans le monde réel. Nous avons produit des échantillons tirés du monde algorithmique et tirés du monde réel. Dans les deux cas, malgré l'existence parfois de substantielles différences, on a trouvé de la structure partout, quelquefois dans une proportion proche de la distribution de motifs que les programmes produisent. La plupart d'entre eux produisent de la structure et seule une petite fraction a l'air aléatoire.

Chapitre 7

Une approche algorithmique de l'information et de la signification

Je vais passer en revue certains sujets pertinents quant au débat philosophique sur l'information, en tenant compte des développements de la théorie algorithmique de l'information (AIT). La thèse que je défendrai est que la signification possède une profondeur, au sens de la profondeur logique de Bennett, et que la probabilité algorithmique peut fournir la stabilité nécessaire pour une définition algorithmique robuste de la signification, une définition prenant en considération l'interprétation et la connaissance propre du destinataire encodées dans l'histoire attachée au message.

7.1 Qu'est-ce que l'information

L'information peut être une pierre angulaire pour interpréter toutes sortes de phénomènes, et constituer la base pour une description des objets. Alors qu'il est légitime d'étudier idées et concepts liés à l'information dans leur sens le plus large, l'usage d'information hors contexte formel peut conduire à des détournements. Il n'est pas inhabituel de tomber sur des études et des volumes consacrés à l'information (au sens le plus large) dans lesquels le débat mathématique ne s'aventure pas au-delà de l'état du domaine où Shannon [[Shannon, 1948](#)] l'a laissé il y a bientôt 60 ans. De récentes percées dans le développement de la théorie de l'information sous sa forme algorithmique – des développements à la fois

théoriques et empiriques dotés d'application dans divers domaines (par ex. [Li and Vitányi, 2008, Li et al., 2001, Li et al., 2004, Zenil et al., 2012a]) – sont souvent négligées dans l'étude sémantique de l'information, et c'est à la philosophie et à la logique (par ex. la logique temporelle épistémique) qu'on a recours pour essayer de tenir compte de ce qu'on appelle le formalisme sémantique de l'information. Comme exemples on peut citer le travail de [Floridi, 2003, Floridi, 2008, Godfrey-Smith and Sterelny, 2007]. Dans le meilleur des cas, la théorie algorithmique de l'information (AIT) n'a pas le poids qu'elle mérite. Ses définitions de base sont parfois rendues approximativement (par ex. une définition incomplète de la profondeur de Bennett [Bennett, 1988] dans [Roederer, 2006]) (p. 25)). Dans [Floridi, 2003] par exemple, la seule référence à l'AIT en tant que contexte formel pour une discussion sur le contenu informatif et la signification, est une référence négative – apparaissant dans la contribution de van Benthem (p. 171 [Floridi, 2003]). On y lit :

Pour moi, l'idée qu'on puisse mesurer le flux informatif sur une dimension, en termes de nombre de bits, ou une autre mesure quelconque, me semble manifestement absurde ...

Je pense que cette position est peu judicieuse. Quand Descartes transforma la notion d'espace en un ensemble infini de nombres ordonnés (coordonnées), il n'a pas privé de tout intérêt le débat et l'étude sur l'espace. Au contraire, il a fait avancer et a étendu le débat philosophique jusqu'à cerner des concepts tels que la dimension et la courbure – ce qui n'aurait pas été possible sans l'intervention cartésienne. Peut-être que ceci répond à la question que se pose Benthem immédiatement après sa remarque ci-dessus (p. 171 [Floridi, 2003]) :

Mais dans la réalité, cette approche quantitative rencontre un succès spectaculaire, souvent beaucoup plus que tout ce que produit mon univers de logique et de sémantique. Pourquoi ?

Accepter un cadre formel tel que la complexité algorithmique, pour étudier le contenu informatif, ne veut pas dire que le débat philosophique sur l'information va se réduire à une discussion sur les nombres impliqués, comme ce ne fut pas le cas pour la philosophie de la géométrie après Descartes.

La thèse fondamentale sur laquelle s'appuie aujourd'hui la théorie de l'information (dérivée des travaux de Shannon) est que l'information peut se réduire à une série de symboles.

Bien qu'il soit possible de discuter légitimement de l'information en se basant sur différentes hypothèses fondamentales, c'est de la pure variation syntaxique, car on peut toujours considérer la théorie de l'information comme achevée par la théorie de la communication de Shannon.

Les débats épistémologiques sont cependant impossibles à concevoir en l'absence d'une notion de sémantique. C'est surtout en logique qu'on a travaillé pour saisir le concept de signification dans un sens formel plus large. Toutefois, rien ou presque n'a été fait pour expliquer la signification en utilisant des modèles purement computationnels, que ce soit en élargissant le travail de Shannon vers l'information, ou en expliquant la signification à la lumière de la fusion initiée par Turing entre information et computation (toute machine peut s'écrire comme entrée d'une autre machine), ou en expliquant la signification à la lumière de développements récents, comme ceux incarnés par la théorie de l'AIT.

L'Entropie de Shannon est définie comme une mesure du contenu informatif moyen associée à un résultat aléatoire. Formellement, posons $d = (p_1, p_2, \dots, p_n)$ comme étant une distribution finie de probabilités discrètes. Ce qui suppose $p_k \geq 0$ pour $k = 1, 2, \dots, n$ et $\sum_{k=1}^n p_k = 1$. L'incertitude concernant un résultat possible doté des probabilités $p_1, p_2, \dots, p_n$ est appelée l'entropie de la distribution P , et on la mesure par $H(P) = H(p_1, p_2, \dots, p_n)$ comme l'a introduit Shannon [Shannon, 1948] et défini par $H(p_1, p_2, \dots, p_n) = \sum_{k=1}^n p_k \log_2 1/p_k$. Cela indique combien de bits sont nécessaires pour coder un message de façon à l'envoyer à travers un canal de communication à capacité minimum. Cela compte combien de symboles différents possède un message, pondérés par leur distribution de probabilité. C'est donc une mesure de la diversité, et pas de l'ordre ni du désordre.

La sémantique concerne le contenu. Les composantes syntaxiques et sémantiques de la théorie de l'information sont concernées toutes deux par l'ordre, la syntaxique surtout grâce au nombre de symboles et à leurs combinaisons, tandis que la sémantique est intimement liée à la structure. La théorie de la computation est le contexte fourni par la théorie algorithmique de l'information pour débattre du concept d'information. Dans ce contexte, la description d'un message est interprétée en terme de programme. Les sections suivantes sont une vue générale des différentes directions formelles vers lesquelles on a développé l'information ces dernières décennies. Elles laissent beaucoup d'espace pour un débat philosophique fructueux, une discussion orientée vers l'information en soi, ainsi que ses relations avec des aspects de la réalité physique.

7.2 Communication, diversité et complexité

La conception de Shannon hérite des pièges de la probabilité. C'est-à-dire qu'on ne peut pas parler du contenu informatif d'une chaîne individuelle. Néanmoins, des erreurs d'interprétation ont poursuivi la mesure de Shannon de l'information, et ceci depuis le début, surtout autour de l'usage du terme entropie, comme Shannon lui-même l'a reconnu. Le problème a été qu'on a pris l'Entropie de Shannon comme une mesure de l'ordre (ou du désordre), comme si c'était une mesure de la complexité (semblable à l'entropie physique dans la thermodynamique classique). Shannon a reconnu que sa théorie est une théorie de la communication et de la transmission, et pas de l'information [Shannon, 1948].

Contrairement à la complexité algorithmique, l'Entropie de Shannon ne prend pas en considération la structure interne du message. Considérons les chaînes 0101010101 et 0100101100. Elles possèdent toutes deux exactement la même Entropie de Shannon, car le nombre d'occurrences de 0 et de 1 est le même dans les deux chaînes. Comme second exemple, considérons la phrase "A quick brown fox jumps over the lazy dog" et sa version emmêlée "rl y feawbojkouq A vdpegsxioz r emunoth". Toutes deux ont la même valeur d'Entropie de Shannon, même si clairement l'une nous parle alors que l'autre n'a pas de sens. A la base, l'Entropie de Shannon dit que, à cause du fait que chacun des messages de ce couple de phrases utilise le même nombre de symboles, on a besoin d'un canal de communication de même taille environ pour les deux membres du couple.

Quand il s'agit de transmettre un message "sensé" dans une langue donnée, Shannon ne voit pas les choses de cette manière, il considère qu'on a affaire à un processus markovien discret (ou un automate fini probabiliste). Il ne s'intéresse donc pas seulement aux probabilités a priori d'apparition des symboles (d'après leur fréquence dans la langue), mais aussi aux probabilités des transitions, donc à la fréquence des digrammes, des trigrammes, etc. Par exemple, après "v" un "d" est très peu probable et après "vd" la probabilité d'un "p" est nulle. Au contraire, après "bro" l'apparition de "wn" a une assez forte probabilité. Il pensait qu'on pouvait obtenir ainsi des approximations de plus en plus plausibles des contraintes réelles de la langue. En fait cette théorie de la grammaire est très insuffisante, pour des raisons montrées par Chomsky.

Que la mesure de Shannon soit facilement calculable en pratique, peut jouer en faveur de son application fréquente et déraisonnable en tant que mesure de complexité. Le fait que la complexité algorithmique ne soit pas

calculable, toutefois, ne signifie pas qu'on ne puisse pas l'approximer et donc l'utiliser et tester.

L'approche de Shannon n'est d'aucune aide pour définir le contenu informatif ou la signification. Par exemple, pensez à un nombre comme π , qu'on pense être normal (ses digits sont également distribués), qui donc a peu ou pas du tout de redondance. Le nombre π n'a pas de motif répétitif (car c'est un nombre irrationnel). En l'absence de structure, pas moyen d'optimiser le canal par lequel le transmettre. Cependant π peut grandement se compresser en se servant d'une de ses courtes formules connues générant tous ses digits, de façon à ce qu'on puisse envoyer la formule plutôt que les digits. Mais ce type d'optimisation est hors de portée de la théorie de la communication de Shannon. Contrairement au traitement de Shannon pour π , on peut penser à π en tant que nombre rempli de sens, car il représente la relation entre toute circonférence et son diamètre. Je vais défendre l'idée que cette signification peut se traiter formellement, en utilisant des concepts de l'AIT, comme approche de ces questions.

Le point de pertinence essentiel pour nous, réalisé par Shannon quand il a formulé sa mesure dans le contexte de la communication, est qu'en pratique un message sans redondance est plus probablement porteur d'information, si on suppose qu'on transmet plus que juste des bits aléatoires. Si quelque chose a l'air aléatoire, on va alors le considérer habituellement comme privé de sens. Pour dire que quelque chose est significatif, cela implique habituellement qu'on puisse arriver à une conclusion en se basant sur lui, d'une manière ou d'une autre. L'information possède un sens seulement si elle a un contexte, une histoire derrière elle. Signifier, dans un monde causal, est l'histoire attachée à un message comme je vais expliquer ensuite.

7.3 Données + un programme est un message + interprétation

Parmi les nombreuses contributions faites par Alan Turing sur la base de son concept d'universalité computationnelle, il y a l'unification des concepts de données et de programme. Les machines de Turing sont un appareillage abstrait et extrêmement simple, manipulant des symboles, qui malgré sa simplicité, peut s'adapter pour simuler la logique de tout calculateur pouvant être construit. Alors qu'on peut être tenté de voir l'entrée (input) d'une machine de Turing comme une donnée, et le tableau de règles de la machine comme son programme, chacun d'eux étant des entités séparées, ils sont en

fait interchangeables, et c'est une conséquence de la computation universelle, comme Turing l'a montré lui-même. Car pour toute machine de Turing M avec entrée s , on peut construire une machine M' avec une entrée vide, de telle sorte que M et M' acceptent le même langage.

Autrement dit, on peut toujours incorporer une donnée comme partie du tableau de règles d'une autre machine. L'identification de quelque chose en tant que donnée ou en tant que programme n'est donc qu'une simple convention par habitude, et non une distinction fondamentale. Ceci est digne d'être remarqué, car bien qu'il puisse sembler qu'un message soit une donnée et qu'une interprétation soit un programme, une partie de l'argumentation contre cela est que le message ne peut se saisir formellement à la fois en termes computationnels et sémantiques, que si l'interprétation fait partie de lui. Donc même si on peut faire une distinction en surface, tout comme on fait une distinction entre un message et son destinataire dans le monde réel, la différence n'est pas essentielle. D'où la soumission de toute chose sous l'intitulé de "message" va nous permettre de définir le concept de signification basée sur la théorie de la computation et de l'information algorithmique. Par exemple, des messages (inputs) pour lesquels une machine de Turing s'arrête peuvent être pris comme des messages ayant été compris. Il est certain que dans le cas d'une machine de Turing universelle, il va y avoir des messages pour lesquels la machine va s'arrêter et d'autres pour lesquels elle ne va pas s'arrêter, une propriété désirable dans la mesure où nous nous occupons de définir la signification d'un message. Ce qui revient à dire que le sens du message n'a pas d'importance quant au fait qu'il y aura toujours des destinataires capables de l'interpréter, et d'autres qui en seront incapables. Que certaines comprennent un message en s'arrêtant (c-à-d conventionnellement) ne veut pas dire, cependant, que toutes le comprennent de la même manière, car chacune peut réagir de façon différente face au message, ce qui est une propriété désirable dans la mesure où on cherche à définir le concept de compréhension subjective. Ces premières conventions simples vont nous permettre d'appliquer plusieurs concepts tirés de la théorie de l'information algorithmique, notamment les concepts de complexité conditionnelle, de probabilité algorithmique et de profondeur logique, de façon à donner du sens et de la robustesse à notre approche algorithmique du concept de signification à partir d'une perspective computationnelle.

7.4 Contenu informatif et signification

En tout cas, la notion d'information selon Shannon laisse clairement entendre que le contenu informatif est subjectif (Shannon lui-même) :

Fréquemment, les messages ont un sens : ils font référence ou sont corrélés à un certain système grâce à certaines entités physiques ou conceptuelles. Ces aspects sémantiques de la communication ne concernent pas le problème d'ingénierie. L'aspect important est que le message concret est un message sélectionné parmi un ensemble de messages possibles [Shannon, 1948].

Subjectif ne signifie pas, cependant, qu'on ne puisse pas définir formellement le contenu de l'information, cela implique seulement qu'on doit inclure une interprétation plausible dans la définition, un point qu'on va explorer dans la prochaine section.

La contribution de Shannon est le point de départ, en ce qu'elle définit le bit en tant qu'unité de base de l'information, comme le font nos meilleures théories actuelles de la complexité informationnelle. Toute série de symboles peut se traduire en une série binaire, préservant par là le contenu original, car elle peut se traduire dans un sens ou dans l'autre, de l'original vers le binaire et vice-versa. La théorie de l'information de Shannon a une approche syntaxique de l'information : si et combien d'information (plutôt que quelle information) est convoyée. Et en tant que phénomène physique : l'idée basique est de rendre le canal de communication plus efficace.

Contrairement à l'Entropie de Shannon, la complexité algorithmique considère les objets individuels indépendamment de toute distribution de probabilités. Par exemple, différents segments initiaux de la suite de Fibonacci ont chacun une entropie différente. Chacun des 10 segments ayant 10 valeurs de plus en plus grandes dans la suite de Fibonacci correspond à la série suivante des valeurs d'Entropie de Shannon : 2.163, 2.926, 3.354, 3.654, 3.88, 4.071, 4.228, 4.36, 4.484, 4.591 – simplement parce que plus les séries de nombres sont longues, plus il faut de bits pour les coder comme telles à travers un canal de communication. Pourtant on pourrait décider que, pour récupérer le même message d'à peu près la même taille, il soit préférable d'envoyer la formule associée à la quantité de nombres de Fibonacci à générer. Cette série peut s'écrire comme $F(n) = F(n - 1) + F(n - 2)$ pour $n = \{3, 4, 5, \dots\}$, $F(1) = 1$, $F(2) = 1$. Laquelle formule possède une complexité algorithmique basse et (presque) constante pour tout segment de la suite de Fibonacci. Cet exemple révèle l'existence d'une différence significative :

deux messages de complexités algorithmiques différentes peuvent avoir la même Entropie de Shannon. L'Entropie de Shannon représente une limite absolue de la meilleure compression possible sans pertes de la complexité algorithmique, c-à-d que $C(s)$ (la complexité algorithmique selon la taille des programmes) ne peut pas être plus petite que $H(s)$ (Entropie de Shannon). Cependant, la suite de Fibonacci, en tant que message, a (presque) la même taille pour la complexité algorithmique, encore une autre caractéristique désirable de la complexité algorithmique (et clairement une qui ne serait pas disponible si le problème était abordé du côté de l'Entropie de Shannon). De plus amples informations sur l'Entropie de Shannon comparée à la complexité algorithmique sont disponibles en [Grunwald and Vitányi, 2004]. On pourrait dire que la signification des valeurs de Fibonacci comme message est la relation qui génère les valeurs de manière récursive et donc sa formule. Il est donc hautement désirable que le contenu d'information soit le même pour n'importe quelle suite des valeurs de Fibonacci qui permettent l'identifier comme la suite générée par cette formule. Donc, une définition de signification basée sur le contenu algorithmique d'une suite est plus robuste qu'une basée sur le contenu d'information classique (Entropie de Shannon).

7.5 Contenu informatif et signification intrinsèque

Comme tentative pour définir le manque de signification, pensons à un bit unique. Un seul bit ne porte aucune information, et il ne peut pas être signifiant s'il n'y a aucun destinataire pour l'interpréter comme quelque chose de plus riche. L'Entropie de Shannon d'un bit isolé est 0, car on ne peut pas établir de canal de communication – aucun message ne peut être encodé ou transmis avec une seule lettre de l'alphabet. C'est intrinsèquement vide de sens car il n'y a pas de contexte. C'est la même chose pour une chaîne de n bits identiques (que ce soit des 0 ou des 1). Pour lui donner un sens, on est obligé de fabriquer une interprétation extérieure, car même si elle transporte un message, il ne peut pas être intrinsèquement très riche, simplement parce qu'il ne peut pas porter beaucoup d'informations. Dans les deux cas, l'Entropie de Shannon et la complexité algorithmique d'une telle chaîne sont très faibles.

A l'autre extrême, une chaîne aléatoire ne peut habituellement pas se considérer comme pleine de sens. Ce qu'on peut dire avec certitude, c'est que quelque chose de significatif devrait donc se trouver entre ces deux extrêmes : aucune information (trivial) ou complet non-sens (aléatoire). La complexité

algorithmique oppose ce qui est simple à ce qui est complexe ou aléatoire, et peut se voir comme une première approximation de la signification, en tant qu'opposant ce qui est significatif dans un sens grossier, à ce qui est vide de sens. Toutefois, la complexité algorithmique seule ne suffit pas à définir la signification, car elle ne peut pas opposer de façon appropriée la complexité à l'aléatoire, mais uniquement opposer la simplicité à la complexité ou l'aléatoire. Ceci parce que la complexité algorithmique associe l'aléatoire au plus haut niveau de complexité. Mais en utilisant la profondeur logique de Bennett [Bennett, 1988] (prenant aussi ses racines dans la complexité algorithmique), on est capable de faire la distinction entre quelque chose qui a l'air organisé et quelque chose qui a l'air aléatoire ou trivial, en introduisant le temps (un paramètre qui semble inévitable dans le déploiement d'un message, ce qui rend raisonnable d'associer cette mesure avec une mesure de complexité physique comme Bennett lui-même le suggère [Adriaans, 2010]). Cette connexion avec la profondeur logique de Bennett signifie qu'un message ne peut pas devenir instantanément significatif, s'il n'est pas dans un contexte approprié.

La complexité algorithmique conditionnelle est définie [Li and Vitányi, 2008] comme la longueur du plus court programme p , pour lequel la machine de Turing universelle U sort s étant donné x , ce qui fait que $C(s|x) = \min\{|p| : U(x, p) = s\}$. A noter que cette dernière définition de la complexité algorithmique est un cas particulier de la complexité algorithmique conditionnelle, quand il n'y a pas de x , c-à-d $C(s) = C(s|\epsilon)$, qui est la complexité algorithmique de s étant donné aucune autre information.

Il est généralement admis que le sens est donné par l'observateur, que le sens est l'interprétation du message par son destinataire. Pour que l'information transmise ait une quelconque valeur sémantique, elle doit d'une certaine manière s'ajouter à la connaissance du receveur. Je prétends que lorsqu'on en vient à élaborer du sens à partir de contenu informatif, les propriétés sémantiques requises peuvent être prises en compte en se servant de concepts tels que la complexité algorithmique conditionnelle et la profondeur logique.

Par exemple, la complexité algorithmique conditionnelle peut définir une mesure de distance entre les messages. Il est clair d'après la définition conditionnelle que, même si une chaîne s est privée de sens, la complexité de s étant donné s est très basse, car la longueur du plus court programme produisant s , avec s comme entrée, est le plus court possible.

7.6 La signification est logiquement profonde

Comme on le sait bien, le problème avec la signification, c'est qu'elle est hautement dépendante du destinataire et de son interprétation. Relier la signification au concept de profondeur logique permet avantageusement de prendre en compte l'histoire et le contexte du message, et donc potentiellement de tenir compte de l'interprétation probable du destinataire. Un message plein de sens (court ou long) contient une longue histoire computationnelle lorsqu'il est groupé avec la computation qui va avec, autrement il aura peu ou pas du tout de sens. D'où la pertinence d'introduire la profondeur logique.

Pensez à un numéro gagnant d'une loterie. Le nombre en lui-même peut être vide de sens pour un destinataire, mais si deux parties ont partagé l'information sur la manière de l'interpréter, l'information partagée à l'avance est intégrée dans l'histoire computationnelle du nombre, et comme telle cette information n'est pas sans lien avec le message ultérieur. Le seul moyen d'interpréter un nombre comme étant le nombre gagnant d'une loterie, est de disposer d'une histoire, et pas seulement une histoire qui relate le nombre comme processus de calcul, mais une histoire qui raconte le processus lui-même. Car gagner un prix n'est pas une question de hasard apparent, mais cela concerne la diffusion d'une information (à la fois le nombre et l'interprétation du nombre), ce n'est donc pas le nombre seul qui représente le contenu et la signification du message (le nombre), mais aussi l'histoire qui lui est attachée.

Il existe aussi des messages qui contiennent l'histoire en eux-mêmes. Si au lieu d'un nombre, on y substitue l'interprétation de ce nombre, le message peut être considéré comme significatif dans son isolement. Mais les deux cas possèdent la même profondeur logique, car ils offrent le même résultat, le même temps de calcul, et sont la conséquence de la même histoire (même si dans le premier cas, une telle histoire peut être rendue par une deuxième étape séparée), et ils ont la même origine, donc la définition semble assez robuste.

Si une machine de Turing mouline en réalisant un gros travail, lorsqu'elle est alimentée avec une entrée aléatoire, donnant ainsi l'impression que cette entrée l'impression qu'elle est pleine de sens, il se peut alors que notre approche ne soit pas suffisamment robuste, car quelque chose qu'on prend pour riche de sens n'est qu'une computation aléatoire. Il y a cependant deux réponses possibles à cette objection : D'un côté, la probabilité d'une machine entreprenant une longue computation par hasard est très basse. Calude et

Stay [Calude and Stay, 2008] ont prouvé que parmi les machines qui s'arrêtent, la plupart vont stopper après peu d'étapes. C'est ce qui se passe pour la plupart des chaînes, ce qui veut dire que la plupart des messages sont vides de sens si à la fois le message et la computation ne sont pas en résonnance entre eux en quelque sorte, ce qui demande une capacité intuitive pour considérer quelque chose comme étant un message riche de sens. D'un autre côté, d'après un raisonnement basé aussi sur des résultats de Calude et Stay [Calude and Stay, 2008], la probabilité algorithmique garantit presque la non-existence de machines de Rute Goldberg (de drôles de machines faisant beaucoup d'effort pour accomplir une tâche triviale), ce qui implique que la quantité de travail réalisée (le nombre d'étapes et la longueur du résultat) venant d'une machine de Turing qui s'arrête, va être proportionnelle au contenu informatif du message d'entrée.

7.7 Une définition algorithmiquement robuste de la signification

La probabilité algorithmique, comme définie par Solomonoff [Solomonoff, 1964] et Levin [Levin, 1974] indique que tout résultat possède une probabilité élevée d'être produit par le(s) plus court(s) programme(s) produisant ce résultat. Autrement dit, les messages significatifs devraient avoir peu de chances d'être interprétés comme tels par hasard.

C'est la probabilité algorithmique qui serait responsable de la robustesse de cette approche algorithmique de la signification. La signification d'un message n'a de sens que pour certains destinataires particuliers (pas pour n'importe lesquels au hasard). Un message possédant du sens pour quelqu'un, peut ne pas avoir de sens pour quelqu'un d'autre – tout-à-fait le genre de propriété qu'on pourrait désirer dans un concept s'attaquant au sens de la signification. C'est ce qui se passe quand certaines machines réagissent à une entrée riche de sens plutôt qu'à une entrée aléatoire. La probabilité algorithmique garantit que la plupart des machines vont s'arrêter presque immédiatement s'il n'y a pas d'histoire computationnelle pour un message donné. Autrement dit, il y a une correspondance entre une entrée riche de sens, la durée de computation et la structure du résultat, étant donné la connexion entre la probabilité algorithmique et la complexité algorithmique.

7.8 Un aléatoire fini (tout comme la signification) se situe dans l'œil de l'observateur

Dans un mouvement qui met en parallèle les erreurs et les abus de la mesure de Shannon comme mesure de complexité, la notion de complexité est fréquemment associée, dans le champ des systèmes complexes, avec le nombre d'éléments interactifs ou le nombre de couches d'un système. Les chercheurs qui pratiquent cette association pourraient continuer à utiliser l'Entropie de Shannon car elle quantifie la distribution des éléments, mais ils devraient être aussi conscients qu'ils ne sont pas en train de mesurer la complexité d'un système ou d'un objet, mais plutôt sa diversité, ce qui peut être quelque chose de tout-à-fait différent (bien que grossièrement lié).

Comme cela a été montré par Stephen Wolfram, ce n'est pas toujours le cas que plus les éléments sont nombreux plus la complexité est grande, ou que plus les couches ou interactions sont nombreuses plus la complexité est grande, car les plus simples des systèmes de calcul sont capables de la plus grande complexité apparente [Wolfram, 2002].

La théorie de l'aléatoire algorithmique ne garantit pas qu'une chaîne de longueur finie ne puisse pas être algorithmiquement compressée. D'ailleurs toute chaîne est garantie d'apparaître comme sous-chaîne (avec une probabilité égale) dans toute série infinie algorithmiquement aléatoire. Mais cela concerne aussi la valeur sémantique de l'AIT, étant donné qu'une chaîne finie possède un sens seulement dans un contexte particulier, en tant que sous-chaîne d'une chaîne plus grande, potentiellement plus longue et essentiellement différente. Donc on peut déclarer une chaîne comme étant apparemment aléatoire seulement tant qu'elle n'apparaît pas comme sous-chaîne incorporée dans une autre chaîne finie ou infinie.

Une chaîne peut être aléatoire à l'échelle d'une série infinie, mais comme toutes les chaînes possibles sont contenues dans une série aléatoire infinie, les blocs aléatoires de chaînes peuvent ne jamais paraître aléatoires. Par exemple, considérons la chaîne suivante générée pseudo-aléatoirement (générée avec le générateur de nombres aléatoires par défaut de *Mathematica*) : 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1. Cette chaîne a bien l'air aléatoire quand on la prend comme un tout, mais elle contient plusieurs blocs qui n'ont pas l'air aléatoires du tout (par ex. 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 de la position 19 à la position 29 ; ou 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 de la position 30 à la position 46),

s'ils sont isolés.

[illegible]

C'est la même chose pour un message. Un message peut être contenu dans un autre message plus long, et concrètement son interprétation peut varier d'un contexte à l'autre. D'après la perspective de la complexité algorithmique (quand elle est interprétée en terme de signification), cette propriété peut se voir comme une indication qu'il ne peut pas y avoir d'interprétation ultime du sens d'un message – ce qui est une autre caractéristique désirable de la complexité algorithmique qui reflète notre intuition du concept de signification.

A l'autre extrême, dans ce contexte algorithmique, il y a le nombre de Chaitin [Chaitin, 1975], qui peut se voir comme véhiculant la plus grande signification possible, car il encode tous les messages possibles sous la forme de réponses à toutes les questions possibles encodées par des machines de Turing. Que l' Ω de Chaitin soit inaccessible en pratique, semble être désirable si on veut éviter une contradiction dans le concept de signification, et aussi à la lumière du fait qu'on ne peut pas s'attendre à coder toutes les significations contenues dans un seul message. Autrement dit, la signifiante totale ou la capture de tous les messages possibles est inatteignable via cette approche algorithmique, car Ω est finalement incalculable. C'est justement ce à quoi on s'attendrait intuitivement pour la signification au sens le plus large.

7.9 Vers un agenda philosophique

Les discussions qui précèdent esquissent un possible agenda pour une philosophie de l'information, dans le contexte de l'état actuel de la théorie algorithmique de l'information. Orientée plus sur le cœur de la théorie, une plus profonde exploration des fondements de l'information algorithmique peut prendre plusieurs directions. Il y a, par exemple, la contribution de Levin [Levin, 1974] à la complexité algorithmique, sous la forme de la demi-mesure éponyme, motivée par le désir de fondamentalement amender la pure défini-

tion de Kolmogorov sur la complexité, à la lumière du fait que l'information doit suivre une loi de conservation non-croissante. C'est une motivation apparemment différente de celle qui se trouve derrière la définition de la complexité algorithmique de Chaitin, dans sa version à préfixes libres.

Il y a aussi les lois de symétrie et d'information mutuelle, découvertes par Gács [Gács, 1974], et Li et Vitányi [Li and Vitányi, 2008], qui restent à explorer, à comprendre et à disséquer philosophiquement.

De plus, il y a les différences subtiles mais importantes, impliquées dans la saisie de l'organisation de l'information, grâce à l'utilisation de l'aléatoire algorithmique, ou bien grâce à l'utilisation de la profondeur logique de Bennett [Bennett, 1988], une question portée à notre attention dans [Delahaye, 2009] par exemple. La motivation se trouvant derrière la formulation de Bennett avec son concept de profondeur logique, a été de saisir la notion de complexité en tenant compte de l'historique d'un objet. Cela a suscité, dans notre approche algorithmique de l'information, l'importante conséquence de pouvoir classer intuitivement des objets physiquement superficiels comme objets de peu de sens.

Il y a aussi la question de la dépendance des définitions par rapport au contexte dans lequel la signification est évaluée (le choix d'une machine universelle de Turing), jusqu'à l'idée d'une constante additive, une question récemment évoquée dans [Delahaye and Zenil, 2007, Delahaye and Zenil, 2012], où on propose l'idée que des choix raisonnables de formalismes computationnels conduisent à de raisonnables évaluations de la complexité [Delahaye and Zenil, 2008]. C'est le problème de trouver un cadre stable pour une évaluation suffisamment robuste du contenu informatif.

Dans [Floridi, 2003], van Benthem souligne un problème de grand intérêt philosophique quand il exprime le désir de comprendre la déraisonnable efficacité (une expression qu'il déclare avoir empruntée) des théories quantitatives de l'information.

Paradoxalement, je serais plutôt concerné par la déraisonnable inefficacité des théories qualitatives de l'information, notamment la complexité algorithmique, étant donné que c'est la déraisonnable efficacité des théories quantitatives de l'information, notamment la notion d'entropie informationnelle de Shannon, qui a conduit les chercheurs à se fourvoyer en l'utilisant dans des cadres de travail où une véritable (et universelle) mesure de la complexité est nécessaire. Les relations entre l'Entropie de Shannon et la complexité de Kolmogorov sont étudiées en détail dans [Grunwald and Vitányi, 2004].

C'est l'inefficacité de la complexité algorithmique qui insuffle au contenu informatif sa plus profonde caractéristique, du fait que sa pleine caractéri-

sation ne peut pas se réaliser effectivement, même si on peut précisément la définir. D'où le riche filon qu'a ouvert van Benthem, ce débat étant potentiellement fertile et d'un grand intérêt, même si jusqu'ici il a été largement ignoré. Je pense que les formulations provisoires des lois sur l'information, ainsi que leur motivations sous-jacentes, devraient être une partie essentielle du débat, sinon le principal centre d'intérêt d'une approche sémantique de l'information, qui est plus qu'intégralement enracinée dans les développements mathématiques actuels.

On peut protester qu'une approche sémantique de l'étude de l'information ne devrait pas se réduire à de l'algorithmie, qui est parfois considérée comme une vision syntaxique-digitale de l'information. Cependant, c'est plutôt une objection bizarre, car la plupart des textes sur l'information partent de la théorie de l'information de Shannon, sans tenir compte de l'étape suivante et sans envisager de réfléchir sur l'état actuel du débat sur l'information, par exemple sur l'AIT. Donc, soit la philosophie de l'information doit prendre une direction complètement différente de celle de Shannon, qui conduit inévitablement à l'état actuel de l'information algorithmique motivant une plus profonde investigation sur le sujet, soit la philosophie doit s'affranchir de tout ce côté algorithmique en le considérant séparé et étranger au sujet. Autrement dit, je ne crois pas qu'il soit très cohérent de se replier sur les travaux de Shannon, en excluant tous les développements suivants du domaine, qui s'appuient entre autres sur Kolmogorov, Chaitin, Solomonoff, Levin, Bennett, Gacs et Landauer. Comme je l'ai souligné dans la section précédente, il existe un agenda légitime concernant ce qui peut s'appeler la branche syntaxique-mécaniste de l'étude de l'Information, qui est paradoxalement je pense, la partie la plus intéressante et fertile de l'investigation sémantique, et qui a été traditionnellement négligée par le courant principal de la Philosophie de l'Information, à quelques exceptions près [Parrochia, 1996, McAllister, 2003].

Peu importe que l'information puisse être considérée comme complète sans prendre en compte les interprétations de l'information quantique. Mais un problème pertinent a été soulevé par Wheeler [Wheeler, 1990], quoique peut-être en se référant à une autre échelle, qui se demande si un observateur est nécessaire pour qu'une information existe, et pour qu'une observation ait un sens. Il n'y a pas d'interprétation universellement acceptée de la mécanique quantique, bien que l'interprétation dite de Copenhague soit considérée comme la principale. Les débats sur le sens de la mécanique quantique et ses implications ne conduisent cependant pas à un consensus. Il est hors de portée de ce chapitre de discuter plus avant de l'approche quantique, autrement que de souligner sa pertinence quant à un débat qui doit être le plus large possible.

7.10 Bases sur lesquelles s’entendre

On devrait s’entendre sur des propriétés fondamentales de l’information, tirées de l’état actuel de l’AIT, qui comme je le défends, peuvent servir de base pour un cadre mathématique dans un débat philosophique. Bien que ce ne soit pas le lieu ici pour débattre des résultats de la théorie de la complexité algorithmique, voici une liste non-exhaustive de certains des points sur lesquels se mettre d’accord, illustrant aussi l’affirmation que la signification peut se saisir, et qu’on peut étudier au moins certaines de ses propriétés. La liste, dans un certain ordre de dérivation logique, est :

- L’unité de base de l’information est le bit, mais l’information reste subjective (Shannon [[Shannon, 1948](#)]).
- La mesure de Shannon sur l’information ne permet pas de saisir le contenu, l’organisation ou la signification, car elle n’est pas une mesure du contenu informatif, ni une mesure de complexité.
- La complexité algorithmique est un cadre objectif et universel pour saisir la structure et l’aléatoire.
- L’aléatoire implique l’impossibilité de l’extraction de l’information (Chaitin [[Chaitin, 1975](#)]).
- La superficialité est vide de sens (Kolmogorov [[Kolmogorov, 1965](#)]).
- L’aléatoire est vide de sens, Bennett [[Bennett, 1988](#)]).
- Il existe de fortes connexions à explorer entre la logique et l’(ir)réversibilité thermodynamique (Bennett [[Bennett, 1987](#)], Fredkin et Toffoli [[Fredkin and Toffoli, 2002](#)]).
- L’information peut se transformer en énergie, et l’énergie en information [Landauer [[Landauer, 1961](#)], Bennet [[Bennett, 1988](#)]).
- L’information suit des lois fondamentales : symétrie, non-croissance, information mutuelle et (ir)réversibilité (Gacs [[Gács, 1974](#)], Zvonkin [[Zvonkin and Levin., 1970](#)], Levin [[Levin, 1974](#)], Bennett [[Bennett, 1987](#)], Landauer [[Landauer, 1961](#)]).
- Matière et information sont profondément connectées, bien que ce soit une question ouverte que de savoir laquelle des deux est la plus fondamentale (Wheeler [[Wheeler, 1990](#)], Feynman [[Feynman et al., 2000](#)], Bennett [[Bennett, 1987](#)], Landauer [[Landauer, 1961](#)], Fredkin et Toffoli [[Fredkin and Toffoli, 2002](#)], Wolfram [[Wolfram, 2002](#)]).

L’information a aussi commencé à jouer un rôle majeur dans l’interprétation de la mécanique quantique [[Lloyd, 2006](#), [Cabello and Joosten, 2011](#)], et son statut est supposé fondamental dans certains modèles de la physique moderne [[Svozil, 1993](#)], comme il l’a été en physique classique, notamment en thermodynamique, et plus récemment en cosmologie [[Bekenstein, 2003](#)].

Nous étudierons plus tard la liste de ses propriétés et les relations de l'information avec la physique.

7.11 Remarques de chapitre

Un langage commun et un accord général sur le cadre semblent être nécessaires. On a soutenu que l'information algorithmique convient pour définir le contenu informatif individuel et pour fournir une caractérisation du concept de signification, en termes de profondeur logique et de probabilité algorithmique. Cette caractérisation computationnelle plutôt formelle ne veut pas dire qu'un débat sur l'information algorithmique devrait être privé d'un intérêt philosophique légitime.

J'ai brièvement attiré l'attention et entamé une discussion sur certaines des questions pouvant se rapporter à une philosophie de l'information algorithmique liée à une définition sémantique de la signification. J'ai défendu l'idée que la complexité algorithmique et la probabilité algorithmique, associées à la profondeur logique de Bennett, pouvaient constituer un cadre computationnel approprié à l'intérieur duquel débattre du contenu informatif et du sens était possible. Le fait que la signification puisse être pleinement formalisée ne signifie pas non plus qu'elle va perdre ses caractéristiques les plus précieuses, telle que la subjectivité face à un destinataire. La richesse de cette dimension subjective peut être saisie computationnellement, comme je l'ai proposé ici auparavant, et peut constituer un autre point de départ pour l'organisation d'une réflexion philosophique tenant compte et couvrant un domaine qui ne peut plus aujourd'hui être tenu à l'écart des débats philosophiques sur l'information.

En résumé, on soutient qu'une condition fondamentale pour formaliser la signification devrait utiliser soit le concept de profondeur logique ou un concept aussi puissant. Ainsi, par exemple, un tableau monochrome (ou aléatoire) ne peut pas avoir de la signification pour un observateur que s'il est dans un contexte profondément riche, par exemple l'histoire de l'artiste ou une histoire associé au tableau. Des idées similaires ont été développées par Jean Paul Delahaye (voir <http://www.scilogs.fr/complexites/>) où il est arrivé à une conclusion proche à la mienne dans ce sens.

7.11.1 Remerciements

L'auteur voudrait remercier Peter Boltue pour ses suggestions, et Selmer Bringsjord pour ses commentaires, objections et vues précieuses sur l'information dans ce chapitre. D'après sa perspective plus purement logique, Selmer et moi sommes en désaccord en plusieurs aspects, mais nous sommes d'accord sur les points de désaccord.

Chapitre 8

Une approche comportementaliste du calcul naturel et un test de programmabilité

Que signifie prétendre qu'un système physique ou naturel calcule ? Une réponse, soutenue ici, est que le calcul correspond à la programmation d'un système pour qu'il se comporte de différentes façons. Ce chapitre offre une contribution sur ce que signifie calculer pour un système physique, en se basant sur cette idée. Il propose une caractérisation comportementale du calcul en terme de mesure de programmabilité, qui reflète la capacité d'un système à réagir à des stimuli extérieurs. La mesure proposée de programmabilité est utile pour classer les calculateurs en termes de complexité algorithmique apparente de leur évolution dans le temps. Je fais quelques propositions spécifiques s'y rapportant et discute de cette approche dans le contexte d'autres approches comportementales, notamment le test de Turing de l'intelligence artificielle. J'anticipe aussi de possibles objections et considère l'applicabilité de ces propositions quand on s'attache à mettre en rapport la computation abstraite et la computation de type naturel.

8.1 Introduction

Face à la question de la computation, on peut être tenté de suivre la position mathématique formelle et invoquer simplement le modèle de Turing. Ce chapitre n'a pas besoin de faire cela, bien que son auteur soit un

inconditionnel de la beauté et de la généralité du modèle universel des machines de Turing, qui, on va le montrer, est aussi un fondement naturel de la computation non-conventionnelle (et de type naturel).

Aujourd’hui, l’étude des limites de la computation a réussi à offrir de bons indices pour répondre à cette question. La frontière entre décidabilité et indécidabilité a donné une intuition essentielle pour la recherche d’une meilleure compréhension de la computation. Toutefois, on peut se demander à quoi s’attendre exactement avec une telle approche, et si d’autres approches alternatives pour comprendre la computation ne pourraient pas s’ajouter à cette connaissance et à l’intuition qu’elle permet, surtout dans les usages modernes de la computation, là où les objets ou événements sont vus comme des computations dans le contexte de la physique.

Une telle approche implique non pas l’étude des systèmes se cachant derrière une limite incalculable (la “limite de Turing”), mais plutôt l’étude de systèmes aux limites extrêmes du calculable, autrement dit l’étude des besoins minimums pour atteindre le calcul universel. Est-il simple ou compliqué de monter une machine qui soit Turing-universelle ? Cette approche bottom-up minimaliste est incarnée par le projet de Wolfram [Wolfram, 2002], dans sa quête de l’étude des programmes simples, un projet initié par Minsky [Minsky, 1967] et auquel plusieurs auteurs ont contribué (voir [Woods and Neary, 2009] pour une excellente analyse). La question sous-jacente concerne alors le degré de pénétration et d’ubiquité de la propriété computationnelle d’universalité dans les systèmes computationnels et naturels. D’après les résultats variés venant des petits systèmes universels de calcul, nous savons maintenant qu’il en faut très peu pour générer de l’universalité, si bien qu’il semblerait qu’il soit plus difficile de concevoir un langage informatique non trivial et non Turing-complet qu’un langage Turing-complet simplement. Donc il semble naturel de croire que la computation et l’universalité ne soient pas réservées aux calculateurs digitaux. Ce chapitre est organisé comme suit. Dans la Section 1, on discute des fondements de la computation naturelle, en prenant comme point de départ le cas de Turing – avec des arguments en rapport avec la computation digitale – en faveur de l’essence désincarnée de la computation naturelle, qui suppose bien un minimum de substrat physique mais le substrat n’est plus important et peu être faible en ressources. En effet, c’est d’abord par la notion d’état abstrait ou computationnel que Turing le conçoit, et à partir de là de machine abstraite caractérisée par sa table d’instructions. Cette table peut être codée par un mot ou un nombre qui pourra servir de donnée à la machine universelle, capable de la décoder et d’imiter le comportement que prescrit cette table. Les opérations physiques (ou un grand partie) sont donc remplacées

par des manipulations symboliques.

8.2 Un fondement classique pour la computation non-conventionnelle

Un compilateur écrit entre des systèmes computationnels, donc une règle de conversion entre symboles et états, est la façon habituelle de prouver, de manière technique, qu'un système est équivalent à un autre en termes de puissance de calcul (donc qu'il calcule). On peut alors légitimement se demander si on a besoin de cet appareillage technique pour définir la computation. Le problème peut s'énoncer avec les mots de M. Conrad *Dans le monde réel, on sait peu ou pas grand-chose sur les opérations primitives ou les symboles d'un système*. Une forte critique de l'idée que des objets naturels (dont l'univers) calculent, est que question et réponse se trouvent vite privées de sens, car il est difficile de voir comment tout système physique ne serait pas computationnel [Putnam, 1988, Searle, 1992]. Un concept que Turing n'a pas proposé (bien qu'il ait suggéré de prendre en compte le pourcentage de gens reconnaissant le succès ou l'échec de son test d'intelligence des machines [Turing, 1950]), mais qui est tout à fait dans l'esprit d'une autre de ses riches contributions (la relativisation de la computation, par la notion de degrés de computation [Turing, 1939]), c'est une métrique de l'intelligence, où échouer et réussir n'est pas l'essentiel, mais plutôt nous dire si on est proche ou éloigné d'un comportement intelligent.

Ce chapitre propose une métrique pour un comportement à la limite, approximatif et asymptotique, non pas pour l'intelligence, mais pour la computation, où on identifie des objets auxquels on peut attribuer un certain degré de computation en se basant sur leur comportement, et surtout en se basant sur la possibilité de les programmer. Cela place donc la programmabilité au cœur de notre définition de la computation, en évitant ainsi le représentationnalisme.

8.2.1 Une approche comportementale de la computation

Parmi les plus importantes contributions de Turing envers l'intelligence artificielle (IA), il y a eu son test sur l'intelligence des machines [Turing, 1950], conçu pour répondre à la question de savoir si les

ordinateurs pouvaient penser. Le test de Turing est une approche comportementaliste pragmatique face au problème de l'intelligence des objets (voir la figure 8.1). Dans l'esprit de Turing, on peut se demander si des objets autres que des calculateurs électroniques, calculent, en particulier des objets naturels et des processus naturels. Cette question conduit finalement à la question plus générale de savoir si l'univers lui-même calcule (connue aussi sous le nom de "pancomputationnalisme"), et si c'est le cas, comment. On a donné certaines réponses spéculatives, mais dans cette présentation, nous allons adopter une approche plus pragmatique et comportementaliste face à cette question, dans l'esprit de l'approche de Turing face à l'intelligence.

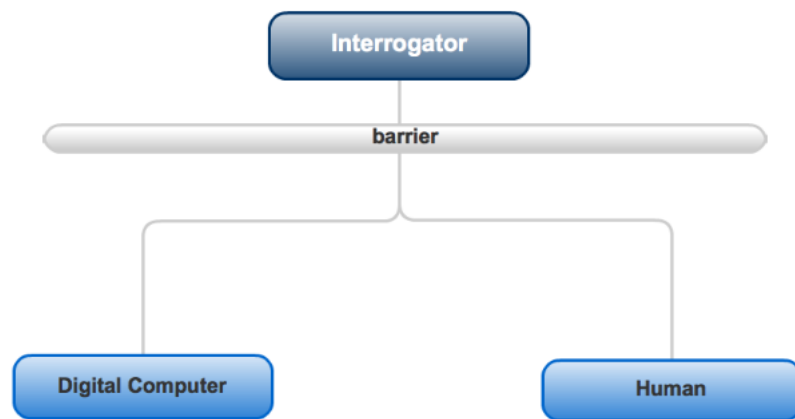


FIGURE 8.1 – Les éléments de base du test de Turing sur l'intelligence.

Quand Alan Turing pensait à l'intelligence artificielle, il croyait "que dans à peu près cinquante ans il sera possible de programmer des ordinateurs pour les faire si bien jouer au jeu de l'imitation qu'un interrogateur moyen n'aurait pas plus de 70 chances sur cent de les identifier correctement après cinq minutes de questionnement... Je crois qu'à la fin du siècle, l'usage des mots et l'opinion de l'homme de la rue auront tellement changé qu'on pourra parler de machines pensantes sans risque d'être contredit". Beaucoup seraient d'accord pour dire que la vision de Turing ne s'est pas exactement réalisée, peut-être à cause de la façon dont s'est modifiée la définition de l'intelligence entretemps, en fait à chaque fois qu'une certaine tâche requérant de l'intelligence a été pleinement exécutée par un ordinateur, du traitement des nombres plus vite qu'un humain jusqu'à faire mieux qu'eux aux échecs, et plus récemment participer à certains jeux plutôt compliqués lors d'émissions de télévision. Je pense qu'on vit un moment où il est finalement devenu

courant de traiter comme calculants des objets autres que les calculateurs humains ou digitaux, et donc je vais appliquer l'inéluctable généralisation du concept de computation au-delà du royaume des ordinateurs, et plus précisément l'étendre aux systèmes naturels. S'il fallait revoir les prétentions de Turing, remplacer "machines pensantes" par objets calculants rendrait la prédiction exacte : "Je crois qu'à la fin du siècle, l'usage des mots et l'opinion de l'homme de la rue auront tellement changé qu'on pourra parler de (toutes sortes d'objets calculants) sans risque d'être contredit."

8.2.2 La computation digitale en tant que computation naturelle

La plus importante contribution de Turing à la science fut sa définition de la computation universelle, elle-même partie intégrante de sa tentative pour mécaniser le concept de machine calculante. Une machine universelle (de Turing) est un appareillage abstrait capable de supporter toute computation pour laquelle on peut écrire une instruction. Plus formellement, étant donné une description fixe des machines de Turing, on dit qu'une machine U est universelle si, pour toute entrée s et toute machine de Turing M , $U(\langle M \rangle, s)$ s'arrête si M s'arrête sur s et donne en résultat $M(s)$; et elle ne s'arrête pas si M ne s'arrête pas sur s (avec $\langle M \rangle$ signifiant M codée en bits pour qu'une machine de Turing U qui s'alimente avec des entrées en binaire, puisse l'accepter). Autrement dit, U est capable de faire tourner toute machine de Turing M avec une entrée s . Le fait qu'on ait besoin de matériel (hardware) et de logiciel (software) indique le besoin d'un substrat programmable pouvant être fabriqué pour calculer quelque chose pour nous, mais la principale contribution de Turing vis-à-vis du concept de computation universelle est que les données et les programmes peuvent se stocker ensemble dans une seule mémoire et sans aucune distinction fondamentale. On peut toujours écrire une machine ayant un effet spécifique, sans entrée, pouvant réaliser un calcul quelconque, et on peut toujours écrire un programme décrivant ce calcul et le prendre comme entrée pour une machine de Turing (universelle), donc, en un sens, il n'y a pas de distinction essentielle entre données et programmes.

Il est clair qu'on peut extraire un type fondamental de calcul naturel à partir du riche concept de computation universelle de Turing. Turing souligne [Turing, 1950] qu'étant donné que le calculateur de Babbage ne se sert pas d'électricité, et parce que tous les calculateurs digitaux et celui de Babbage sont fondamentalement équivalents d'une certaine façon, l'électricité ne peut pas être une propriété fondamentale de la computation. Ni en être le

support. Autrement dit, l'universalité de Turing désincarne la computation, en la découplant de tout substrat physique. Cela ne signifie pas qu'on puisse produire des computations sans l'aide d'éléments physiques, mais signifie plutôt que la nature des éléments physiques n'est pas importante, sauf pour la question (importante) des ressources (capacités, vitesses). Un programmeur utilise de l'espace mémoire et des cycles de CPU dans un ordinateur standard pour réaliser une computation, mais ce n'est aucunement une indication que la computation nécessite un ordinateur, c'est seulement qu'il faut un substrat. Le comportement du substrat est la propriété sous-jacente qui fait de quelque chose une computation.

La principale différence entre un calculateur électronique digital et un système naturel pouvant calculer, c'est que le premier a été conçu pour cela, et donc on peut facilement identifier tous ses éléments et avoir recours à eux quand on établit la définition d'une computation. Pour les systèmes naturels, cependant, il y a peu d'espoir que, même si leurs éléments ont été identifiés, on puisse définir leur statut de façon à saisir toutes leurs convolutions suffisamment bien pour établir qu'ils possèdent une certaine propriété de computation. Cette situation n'est pas si différente de l'indécidabilité du problème de l'arrêt, mais elle est en quelque sorte plus générale. Pour la computation digitale, l'indécidabilité du problème de l'arrêt signifie que si on veut savoir si une computation va finir par s'arrêter, il n'y aura pas d'autre option que de la lancer et voir ce qui se passe (peut-être pendant un temps infini). Dans les systèmes naturels, le problème de l'arrêt se rapproche du problème de l'accessibilité, qui est la question de savoir si un système va avoir accès à une certaine configuration. On peut réduire le problème de l'arrêt à celui de l'accessibilité, et cela prouve que ce dernier est lui aussi indécidable (si on pouvait le résoudre, cela résoudrait le problème de l'arrêt). Les problèmes d'arrêt et d'accessibilité sont fortement comportementaux et subjectifs par nature, car le comportement d'un système doit être déterminé en patientant, en étant témoin, et en l'enregistrant de façon à ce qu'il puisse être compris rétrospectivement. Si, pour des machines de Turing M , la fonction que calcule M ne peut pas être trouvée dans la plupart des cas, il y a peu d'espoir de pouvoir trouver ou même définir la fonction d'un système naturel. Il faut donc cesser d'essayer de définir la computation pour les systèmes naturels en utilisant des éléments comme les états et les fonctions.

Nous savons que des systèmes que jamais personne n'a conçu comme calculateurs sont capables de réaliser de la computation universelle, par exemple la Règle 110 de Wolfram [Wolfram, 2002, Cook, 2004] (appartenant à l'espace des règles de ce qu'on appelle les automates cellulaires élémentaires [Wolfram, 2002]), et que cette règle, comme d'autres systèmes remar-

quablement simples, est capable de computation universelle (par exemple le Jeu de la Vie de Conway [Li and Vitányi, 2008] ou la fourmi de Langton [Langton, 1986]). On peut dire de ces systèmes qu'ils sont littéralement apparus de manière physique, sans avoir été conçus délibérément. Il n'y a cependant pas d'accord universel sur la définition de ce qu'un calculateur doit ou ne doit pas être, ou sur ce que peut être exactement une computation, même si computation/calcul et calculateurs sont des choses bien saisies intuitivement.

Maintenant, on aimerait bien avoir un concept de computation lié à la fois aux phénomènes naturels et aux phénomènes physiques, qu'on puisse mesurer et sur lequel on puisse s'appuyer. Nous voulons une métrique de la computation qui nous permette d'identifier ce qu'est un calculateur et ce qui n'en est pas un. On veut pouvoir distinguer ce qui calcule de ce qui ne calcule pas. Et on désire disposer d'une métrique utilisable.

8.3 Une approche inspirée du test de Turing, pour la computation

Comme pour le test d'intelligence de Turing, où il faut accepter que les humains pensent pour qu'il ait un sens, le lecteur doit d'abord accepter que la computation digitale se produit dans la nature et que la nature est capable de computation digitale, même si seuls les calculateurs digitaux construits par les humains l'ont été dans ce but. Le comportement humain est au test de Turing ce qu'est la computation digitale à cette approche comportementale de la computation naturelle. L'argument peut être rendu plus succinct ainsi : les calculateurs électroniques calculent, les calculateurs électroniques sont des objets physiques de la nature (ou construits à partir de la nature avec de ressources naturels), les objets physiques font partie de l'univers, une partie de l'univers est donc évidemment capable de computation. On peut voir les ordinateurs comme le résultat de la reprogrammation d'une partie de l'univers pour faire en sorte qu'elle calcule ce qu'on veut qu'elle calcule. Cela veut dire que la question n'est pas exactement de savoir si l'univers est capable de computation digitale mais plutôt de savoir si l'univers se contente de ne réaliser que de la computation, et si c'est le cas, quel type de computation. J'ai l'intention de fournir une définition comportementale de la computation, qui admette une plus large définition de la notion de 'computation'. A noter que je suis en train de remplacer la question de savoir si un système est capable de computation digitale par la question de savoir si un système

peut se comporter comme un calculateur digital, et si un calculateur digital peut produire le comportement d'un système naturel. Ainsi, dans ce sens-là, l'approche reste classique, mais intentionnellement neutre par rapport au problème ontologique. A noter aussi encore la ressemblance avec l'approche de Turing vis-à-vis de l'intelligence des machines. Turing a choisi de parfois parler "d'imitation" au lieu de "comportement." Cependant "imitation" semble impliquer des connotations d'intentionnalité (voir la sous-section 8.5.2), et je ne suis pas très à l'aise quant à suggérer qu'un système naturel puisse avoir une volonté de, ou puisse imiter intentionnellement un autre système, surtout s'il est forcé de le faire artificiellement (bien que l'imitation soit chose assez courante dans la nature où, par exemple, certains animaux miment le comportement d'autres animaux pour éviter de devenir une proie).

Pour donner sens au terme "computation" dans les contextes qui m'intéressent (les visions modernes de la physique), je propose une notion comportementale de la computation de type naturel (dans le même esprit que le néologisme "computation de type physique" [Margolus, 1984, Sutner, 2011]) qui soit compatible avec la computation digitale tout en restant signifiante dans des contextes plus larges, et qui soit indépendante des représentations et des supports possibles. Cela va nécessiter une mesure du degré de programmabilité d'un système basée sur un index de compressibilité qui va finalement trouver ses racines dans le concept de complexité algorithmique. Je pose la question de savoir si deux computations sont les mêmes lorsqu'elles se ressemblent, et j'essaye d'y répondre à l'aide d'un outil spécifique ayant le pouvoir de saisir une notion qualitative du comportement.

En [Cronin et al., 2006], une approche similaire est audacieusement présentée, cette fois au sujet de la vie, mais dans le même esprit que le test de Turing. L'idée est de reconnaître un système vivant grâce à la façon dont il se comporte et communique à travers les signaux transmis entre cellules biologiques. Cette approche utilise un interrogateur biologique pour demander non pas ce qu'est la vie, mais plutôt quand une cellule artificielle peut être qualifiée de vivante.

L'approche comportementale s'empare du concept désincarné de computation universelle posé par Turing, au sujet d'une computation indépendante du substrat et le pousse à ses limites logiques, la question centrale étant de savoir si on peut programmer un système pour qu'il se comporte de façon voulue. Ceci est encore proche du test de Turing où l'interrogateur ne peut pas directement voir la personne qui répond, car l'intelligence n'est pas une propriété qui force son possesseur à avoir une "peau" (selon les mots de Turing lui-même [Turing, 1950]), par exemple, ou à être humain dans ce cas (l'approche de Turing), comme la computation ne force pas à utiliser de l'élec-

tricité, ou à être un ordinateur digital dans ce cas (cette approche). Cette approche elle-même basée sur jusqu'où un système peut être programmé, nous dit jusqu'à quel degré un système donné ressemble à un ordinateur. Comme interrogateur, nous allons utiliser un algorithme de compression sans perte manifestant les propriétés d'un observateur, telles que des ressources finies et un certain type de subjectivité. Comme suggéré par Sutner [Sutner, 2011], il est raisonnable de demander que toute définition de la computation au sens large, plutôt que d'être une description purement logique (par ex. selon les termes de la théorie de la récursivité), doive saisir le sens de ce que peut être une computation physique. Sutner ajoute "Un système physique n'est pas intrinsèquement un ordinateur, c'est plutôt qu'il faut interpréter certaines caractéristiques du système physique comme représentant une computation." Cela oblige Sutner à tenir compte de l'observateur et de l'acte d'interprétation d'un système physique.

	Test de Turing pour l'intelligence [Turing, 1950]	Test de Turing pour la vie [Cronin et al., 2006]	Test de Turing pour la computation
Propriété imitée	Pensée	Fonctions cellulaires	Programmabilité
Sujets en question	Machines calculantes	Cellules biologiques	Calculateurs digitaux et artificielles
Incarnation de la propriété	Intelligence humaine	Vie biologique (métabolisme, évolution, etc)	Tout objet
Mécanismes de démonstration	Questions/réponses via un langage naturel	Questions/réponses via un langage physicochimique (potentiels chimiques, transmission mécanique signalétique, etc)	Evaluation comportementale (sensibilité aux stimuli extérieurs, différences de comportement, etc) via un algorithme de compression sans perte

TABLE 8.1 – Comparaison entre les tests de Turing pour l'intelligence, la vie, et la computation.

A maints égards, ce compte-rendu sur la computation peut se déduire de la négation de l'une des conditions de Piccinini (ce qui ne convient pas, ne calcule pas) [Piccinini, 2008] qui est dogmatique d'après moi, et s'oppose à l'extension de la notion de computation naturelle. Un exemple de computation naturelle est celle qui utilise comme substrate le *Physarum polycephalum*, un organisme unicellulaire capable de faire de calculs et que peut résoudre certain type de problèmes algorithmiques (par exemple, des labyrinthes) [Adamatzky, 2015].

Parmi les choses que Piccinini exclut comme objets pouvant calculer, il y a les systèmes planétaires, les ouragans et les systèmes digestifs. En fait, Piccinini lui-même semble avoir certaines difficultés ([Piccinini, 2008], p.508) pour justifier comment un système digestif est non-computationnel. Car dans la mesure où on peut faire un juste compte-rendu mécaniste d'un système

digestif, cela voudrait dire qu'il possède précisément les types de propriétés et de composants pris en compte pour déterminer si un système équivaut à un calculateur ou non. Je vais défendre l'idée qu'on n'a pas besoin, au niveau axiomatique, d'exclure de tels systèmes de la question de savoir s'ils calculent ou non. Je ne vais pas chercher à revendiquer qu'un système de ce genre calcule ou non, car l'approche proposée ci-dessus est avant tout une approche pragmatique conçue pour avoir des applications (elle fut en fait d'abord développée comme un outil de recherche des propriétés dynamiques de programmes informatiques, et non pas initialement en vue d'une contribution physiologique).

D'un autre côté, le point de vue comportemental défendu ici ne satisfait pas du tout la 2ème (ce qui ne convient pas ne calcule pas) et 3ème conditions de Piccinini (indépendance de l'observateur). Les conditions d'explication) et de taxonomie sont au cœur de la proposition de Piccinini, qu'à mon avis connectent *programmabilité* et *calcul* (computation), et fournissant un système gradué basé sur le comportement. La proposition 4 (erreur de calcul) de Piccinini ne semble pas très pertinente pour cette proposition, et même si c'était le cas, pour lui cette caractéristique ne semble pas essentielle pour la computation, car il est difficile de voir comment un système computationnel peut faire des erreurs de calcul autrement que dans les yeux de l'observateur. En effet, Piccinini lui-même voit cela comme posant un problème dans un compte-rendu de computation, car cela viole la condition 1 (ce qui convient calcul). En fait, de faibles erreurs de calcul (c-à-d dépendantes de l'observateur) sont partout dans la nature ; je pense que la nature manifeste amplement ce type de "mauvais calculs." En résumé, je rejette catégoriquement les condition 2 et 3 (ce qui convient calcule et indépendance de l'observateur), je satisfais aux conditions 1, 4, 5 et plutôt 6, surtout 2 et 5 pour lesquelles je pense que notre proposition excelle. Et concernant la condition 4, je reste neutre, sans dire non-convaincu, bien que je puisse admettre une forme de mauvais calculs faibles, qui seraient une computation n'allant pas dans la direction attendue par l'observateur. Et aussi notre approche permet une taxonomie de la computation. En effet, par contre, mon approche amène à parler de computation aussi chaque fois que l'évolution d'un système se fait selon une loi (physique), ce qui est la majeure contribution du nouveau domaine de la *computation naturelle* à la discussion de ce qui est ou devrait être un "computer" ou un calcul.

8.3.1 La complexité algorithmique comme mesure approximative de la programmabilité

La relation traditionnelle entre comportement et computation a tendance à expliquer le comportement comme une computation [Hodgkin and Huxley, 1952], ou à expliquer la computation comme une simulation de l'activité cérébrale [McCulloch and Pitts, 1943], mais cet auteur n'a pas connaissance des explorations faites en direction d'une explication de la computation en tant que comportement.

Ce chapitre propose une définition comportementale alternative de la computation, basée sur la capacité d'un système à réagir à l'environnement – l'entrée (input) – reflétée dans une mesure de programmabilité. Cela va se faire en utilisant un coefficient de transition de phase, auparavant défini en essayant de caractériser l'évolution d'automates cellulaires et autres systèmes. Ce coefficient de transition mesure la sensibilité d'un système à des stimuli extérieurs, et va servir à définir si un système est susceptible d'être programmé (efficacement), dans le contexte de la définition d'une computation de type naturel. L'observateur de Turing est remplacé par un algorithme de compression sans perte, possédant des qualités subjectives tout comme un observateur standard, au sens où il ne peut que partiellement “voir” les régularités d'une donnée, étant donné que l'algorithme de compression à l'efficacité absolue n'a pas d'existence. L'algorithme de compression va regarder l'évolution d'un système et va déterminer, en alimentant le système avec différentes conditions initiales, si celui-ci réagit aux stimuli extérieurs. La version compressée de l'évolution d'un système est une approximation de sa complexité algorithmique (K).

C'est donc la longueur du plus court programme p qui produit la chaîne s en tournant sur une machine universelle de Turing T [Kolmogorov, 1965, Chaitin, 2001]. Un inconvénient technique de K , en tant que fonction prenant s comme la longueur du plus court programme produisant s , est sa non-calculabilité, démontrée par réduction au problème de l'arrêt. Autrement dit, il n'existe aucun programme qui prenne une chaîne s comme entrée, et produise l'entier $K(s)$ en sortie. D'habitude, on considère cela comme un problème majeur, mais il faut bien s'attendre à ce qu'une mesure universelle de complexité possède une telle propriété. Cette mesure fut conçue initialement pour définir l'aléatoire, et elle est aujourd'hui acceptée comme étant une mesure mathématique objective de la complexité, entre autres raisons par ce qu'elle a été prouvée comme robuste mathématiquement (en ce qu'elle représente la convergence de plusieurs définitions indépendantes). La théorie mathématique de l'aléatoire a prouvé que les propriétés d'un objet

aléatoire pouvaient être saisies par des mesures non-calculables. On peut, par exemple, approcher K en utilisant des algorithmes de compression sans pertes, qui détectent les régularités pour comprimer les données. La valeur de cette méthode par compressibilité est que la compression d'une chaîne en tant qu'approximation de K est un test suffisant de non-aléatoire. Si le plus court programme produisant s est plus grand que s , que la longueur de s , alors s est considérée comme aléatoire.

En se basant sur les principes de la complexité algorithmique, on peut utiliser le résultat des algorithmes de compression ayant été appliqués à l'évolution d'un système pour caractériser le comportement du système en question [Zenil, 2010], en comparant ce résultat avec l'évolution non compressée du système. Si l'évolution est trop aléatoire, la version compressée ne devrait pas être beaucoup plus courte que la longueur de l'évolution originale. Il est clairement possible de caractériser des systèmes par leur comportement [Zenil, 2010] : s'ils sont compressibles, ils sont simples, sinon ils sont complexes (d'apparence aléatoire). On peut pousser plus loin cette approche et s'en servir pour détecter des transitions de phase, comme [Zenil, 2010] le montre, car il est possible de détecter les différences entre les versions compressées du comportement d'un système pour différentes configurations de conditions initiales. Cette mesure secondaire nous permet de caractériser les systèmes sur leur sensibilité à l'environnement : plus ils sont sensibles, plus grande sera la variation en longueur des évolutions compressées. Une classification place en premier des systèmes pouvant être considérés à la fois comme étant des porteurs efficaces d'information et à la fois hautement programmables, étant donné qu'ils réagissent promptement à des perturbations dans leurs entrées. Les systèmes trop perturbables, cependant, ne montrent pas de phase de transition et sont regroupés comme porteurs inefficaces d'information. L'efficacité requise consiste à éviter ce qu'on appelle les engluements (tar pits) de Turing, en l'occurrence des systèmes capables de computations universelles mais très difficiles à programmer en pratique. Cela signifie qu'il y a une différence entre ce qui peut se faire en principe et la capacité pratique d'un système à réaliser une tâche. Cette approche est donc sensible aux aspects pratiques de la programmation d'un système, plutôt qu'à ses capacités potentielles théoriques à être programmé. Que se passe-t-il si, au lieu d'essayer de tirer une ligne claire entre ce qui est et ce qui n'est pas un calculateur, on essayait de définir une mesure de "capacité à calculer" ? Je propose l'approche suivante comme une première approximation de programmabilité.

Prenons C comme étant une approximation de K (étant donné que K n'est pas calculable) obtenue par un moyen quelconque, par exemple en utilisant des algorithmes de compression sans perte, ou bien par la technique

du théorème du codage qu'on a présenté en [Delahaye and Zenil, 2012]. Définissons la fonction f comme la variabilité d'un système M , résultant de l'adaptation d'une courbe ϕ (par régression linéaire) à l'ensemble de points produits par différents parcours de plus en plus longs en durée t' (pour n fixé) jusqu'à un temps donné t , par rapport à la somme des différences de longueur des approximations de la complexité de Kolmogorov (C) d'un système M pour les entrées $i, j, j \in \{1, \dots, n\} \in E$, divisé par $t(n-1)$ (pour la seule raison de normaliser la mesure par le "volume" des systèmes, ainsi on peut grossièrement comparer différents systèmes pour différents n et différents t). Avec E étant une énumération des entrées initiales pour M . L'expression suivante est une tentative de saisir plus formellement t longueurs compressées de M pour différentes conditions initiales i, j . $M_t(i)$ est un système M tournant pendant un temps t et avec une configuration initiale d'entrée i . A la limite, C_t^n capture le comportement de M_t pour un temps $t \rightarrow \infty$, mais la valeur de C_t^n dépend des choix de t et de n (on fait parfois référence à C en supposant un certain t et un certain n), donc on peut seulement tenter de saisir un certain comportement moyen ou asymptotique, s'il existe (car aucune convergence n'est garantie). Cependant, C_t^n est indicateur du degré de programmabilité d'un système M par rapport à ses stimuli extérieurs (les entrées i). Plus grandes les dérivées, plus grande la variation de M , et donc la possibilité de programmer M pour réaliser une tâche ou transmettre de l'information selon un taux saisi par C_t^n lui-même (qui permet en l'occurrence de savoir si, pour un petit ensemble de configurations initiales, M produit un changement unique significatif ou bien des changements incrémentés). Maintenant, la seconde étape est de définir la mesure asymptotique, qui est la dérivée de f par rapport au temps, en tant que programmabilité d'un système (première définition de base) :

$$C_t^n(M) = \frac{(\delta f(M, t, n))}{\delta t} \quad (8.1)$$

Par exemple, comme cela est montré en [Zenil, 2010], certaines règles d'automates cellulaires élémentaires, qui sont hautement sensibles aux conditions initiales et présentent des transitions de phase qui provoquent des changements spectaculaires et qualitatifs de leur comportement lorsque qu'elles démarrent à partir de configurations initiales différentes, peuvent se caractériser par ces propriétés qualitatives. On a entrepris en [Zenil, 2012d] une investigation plus poussée de la relation entre ce coefficient de transition et les capacités de computation de centaines de machines de Turing universelles connues. On va se retenir de faire des évaluations exactes de C pour éviter de distraire le lecteur avec des approximations numériques qui peuvent dé-

tourner l'attention du but particulier de ce chapitre. D'autres calculs sont été proposés en [Zenil, 2014b] et [Zenil, 2012e].

8.3.2 Une approche comportementale de la computation

Les définitions suivantes sont les premières approches de définitions liées au comportement qualitatif de systèmes computationnels :

Variabilité : (le nombre d'évolutions différentes possibles d'un système) : Prenons $U_1, U_2, \dots$ comme étant une énumération des entrées pour un système M . Nous sommes intéressés par le degré de différences entre $M(U_i)$ et $M(U_j)$, en particulier la différence maximum.

Programmabilité : La capacité d'un système à changer, à réagir à des stimuli extérieurs (input/entrées) en altérant son comportement. La programmabilité est donc une combinaison de variabilité et de contrôle externe.

Universalité computationnelle : Programmabilité maximum.

Programmabilité efficace : Les changements maximum de variabilité atteints en un temps polynomial (de degré polynomial faible).

Computation universelle efficace : Universalité avec des variations mesurables, détectées dans un temps polynomial (de degré polynomial faible).

A noter la grande proximité de cette approche avec le test de Turing pour l'intelligence. C'est une sorte de généralisation du test de Turing : la computation est ce qui se comporte comme telle, et c'est le cas si on peut la programmer. S'en suivent les déclarations suivantes (un papier technique avec des définitions formelles est en préparation) :

- Un système U est capable de computation si $C_t^n(U) > 0$ pour $t, n > 0$.
- Un 0-calculateur n'est pas un calculateur, intuitivement dans aucun sens, car il n'est capable de supporter aucun calcul.
- Un système capable de computation universelle (de Turing) possède une valeur limite de C différente de zéro (voir [Zenil, 2012d]). (Une valeur C différente de 0 n'implique toutefois pas une universalité de Turing.)
- Un système U capable d'universalité computationnelle de Turing, converge asymptotiquement vers $\lim C_t^n(U) = 1$ pour $t, n \rightarrow \infty$.

L'utilisation d'un algorithme généraliste de compression sans perte est comparable au rôle joué par un interrogateur dans le test de Turing (voir figure 8.2). Pour l'algorithme de compression, peu importe le support de

computation tant qu'il peut se représenter sous une forme pouvant servir d'entrée quand l'algorithme de compression est lancé. D'un autre côté, un algorithme de compression est associé à des ressources, au sens où il ne peut pas implémenter dans un temps fini tous les tests pouvant effectivement détecter toutes les régularités possibles d'une donnée. Ce qui signifie que l'algorithme est, en quelque sorte, subjectif ; il va d'abord s'occuper de ce qui va le frapper comme structures les plus évidentes à utiliser pour compresser les données. Néanmoins, l'algorithme va le faire d'une façon sophistiquée, avec une plus grande probabilité de succès qu'un compresseur humain, car il est systématique et implémente des méthodes générales. Les algorithmes de compression sans perte peuvent aussi se régler pour tourner plus longtemps afin d'essayer plus de méthodes de compression, tout comme un observateur humain pourrait concevoir plus de méthodes de compression en disposant de plus de temps.

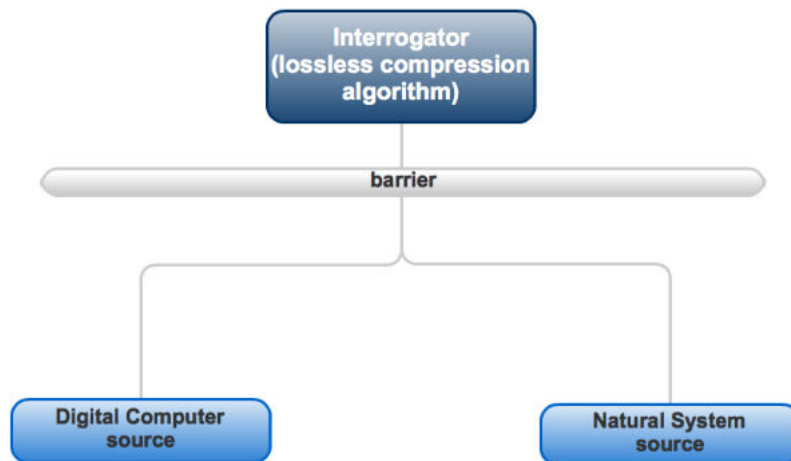


FIGURE 8.2 – Inspirée du test de Turing, l’approche de la question de la computation en tant que test comportemental, entrepris par un algorithme de compression sans perte jouant le rôle de l’évaluateur de réponse. A noter que le système naturel peut être un humain ou n’importe quoi d’autre.

Donc un système S est alimenté avec une entrée aléatoire i (une “question”) et l’algorithme de compression sans perte évalue la réaction du système (puis transforme l’entrée i en une valeur numérique $C(S(i))$, la longueur compressée de $S(i)$ en utilisant l’algorithme de compression C). Tout comme le feraient des observateurs pour apprécier une régularité (ou évaluer une réponse), différents algorithmes de compression peuvent récupérer différentes longueurs de compression de $S(i)$, car ils peuvent avoir différentes façons de

compresser. Cette valeur compressée n'est pas complètement arbitraire, car il y a une certaine objectivité, en un sens forte et désirable. Il en est ainsi car la compression sans perte est un test suffisant du non-aléatoire, voulant dire que si un algorithme C de compression sans perte est capable de compresser $S(i)$, alors la complexité de Kolmogorov de valeur $K(S(i))$ ne peut pas être plus grande que $(C(S(i)))$. D'un autre côté, aucun algorithme C' ne peut compresser $S(i)$ de façon à ce que $(C'(S(i))) < K(S)$, par définition de K , donc les valeurs apportées par un algorithme de compression C ne sont pas complètement arbitraires (ou subjectives).

On peut critiquer la configuration décrite en 8.3 à cause de l'absence d'un véritable questionneur, étant donné que l'algorithme de compression évalue les réponses mais ne formule pas les questions, cela revenant à dire que le test, contrairement à celui de Turing, n'est pas auto-suffisant. C'est une très bonne remarque légitime, mais grâce à Turing, elle n'est pas très fondée. Car, grâce à lui, on sait qu'un système S avec une entrée i peut se réécrire $S'(< S >, i)$, qui est juste un nouvel S' encodant S avec l'entrée i . On peut faire cela pas seulement pour une seule entrée, mais pour tout nombre d'entrées, même un nombre infini d'entrées comme dans une énumération. Prenons E comme étant une énumération pour S , et p_E le programme produisant E (on sait que ce programme existe par définition). Prenons ensuite $S'(< S >, < p_E >)$, tel que S' se comporte comme S et utilise p_E pour alimenter S avec un nombre infini d'entrées (tout comme S pour i , S' peut ne pas s'arrêter). Donc, dans un certain sens, le système est solidement neutre quant à recevoir toutes ses questions d'un coup ou non.

8.4 Une taxonomie de la computation

La mesure proposée en 8.3.1 peut servir à définir dynamiquement la computation en se basant sur le degré de programmabilité d'un système. L'avantage à utiliser le coefficient de transition C est qu'il est indifférent par rapport aux états internes, au formalisme ou à l'architecture d'un calculateur ou d'un modèle de calculateur ; on n'a même pas besoin de signifier si une machine est analogique ou digitale, ou quelle est sa puissance de calcul maximum. C'est uniquement basé sur le comportement du système en question. Cela nous permet de caractériser a minima le concept de computation, sur la base du seul comportement. Maintenant, on peut attribuer la propriété de computation à des objets naturels et physiques, et ainsi arriver à mesurer une Computation de type naturel, et faire la distinction entre les objets physiques grâce à leurs attributs computationnels dépendant de leur programmabilité.

Notre proposition a beaucoup de ressemblances avec l'approche mécaniste de Piccinini, qui fournit une hiérarchie des objets calculants. Mais tandis qu'il met les ordinateurs et les calculateurs (spécifiquement dédiés) dans des catégories différentes, je ne vois pas de raisons essentielles de faire cela. Il place le concept de programmabilité au centre de la discussion, comme je le fais, mais tout dans notre approche est très différent. Son approche mécaniste ne me semble pas vraiment adaptée pour la computation naturelle. A un niveau plus fondamental, l'approche de Piccinini diffère de notre approche au sens où il attribue de l'importance à l'implémentation physique d'une computation et à ses composants physiques, alors qu'on y accorde peu d'intérêt ici. Contrairement à Piccinini, je ne pense pas que la propriété de pouvoir calculer est une caractéristique objective d'un système. On peut définir un programme qui transforme un calculateur généraliste en un calculateur spécialement dédié. Ce n'est pas une définition inconnue, car dans le contexte de l'informatique, une computation peut se voir comme l'évolution subie par un système quand on fait tourner un programme. Cependant, bien qu'intéressante en elle-même, et non sans une certaine affinité avec notre approche, cette voie passant par la définition d'un ordinateur généraliste est une voie indirecte pour définir la computation. Car c'est s'engager à définir l'universalité computationnelle avant d'avoir pu avancer sur la définition de quelque chose de plus basique, quelque chose qui ne devrait normalement pas dépendre d'un concept si puissant (et probablement plus difficile à définir). La computation universelle est sans doute la caractéristique la plus importante de la computation, mais chaque fois qu'on essaye de définir la computation par rapport à une computation universelle, on tombe dans une déclaration circulaire [une computation est une computation universelle (de Turing)], conduisant simplement par là à une version de la thèse CT (Church-Turing). Comme Piccinini le suggère en [Piccinini, 2008], un calculateur universel de Turing, et donc aussi un être humain, peuvent faire plus que suivre un algorithme. Ils peuvent suivre n'importe quel algorithme, pourvu qu'on le leur donne sous forme d'instructions. "Plus généralement, un humain peut être instruit pour réaliser la même activité (par ex. faire du tricot ou jouer du piano) de nombreuses manières différentes. Toute machine pouvant être finalement motivée pour fournir différentes formes de résultats, peut être qualifiée de "programmable." Autrement dit, "être programmable" signifie être motivable pour réaliser des séries relativement longues d'opérations différentes, selon différentes façons dépendantes de la motivation prescrite."

Si des choses de la vie de tous les jours, comme un frigo ou une lampe, peuvent être jugées comme computationnelles, alors il est difficile de voir comment n'importe quel système physique ne serait pas aussi jugé compu-

tationnel (ceci est en rapport avec la théorème de la réalisation de Putnam, voir la sous-section 8.5.2). On peut maintenant pleinement poser la question de savoir si un frigo ou une lampe sont ou ne sont pas des calculateurs, sans faire tomber dans le trivial la question elle-même ou les réponses possibles. L’output résultant d’une lampe, par exemple, peut se décrire par deux comportements différents (dans ce cas, traditionnellement identifiés comme des états), en l’occurrence allumée et éteinte, qui sont déclenchés par une entrée extérieure (via un interrupteur). Même si la lampe peut être considérée comme réagissant à des stimuli extérieurs, elle est très limitée dans son comportement, et l’espace de ses configurations initiales est fini et petit (il n’y a que deux configurations initiales possibles). Donc la pente des différences de l’évolution comportementale dans le temps est très proche de 0. Une lampe est donc un calculateur très limité avec une valeur de C très proche de 0. Si on voulait exclure les lampes et les frigos de la catégorie des appareillages calculants, il suffirait de définir un seuil au-delà duquel un système pourrait être qualifié de calculateur et en deçà duquel il ne serait pas un calculateur. Avec une définition de programmabilité, on peut s’attendre à pouvoir construire une hiérarchie des objets calculants (voir Tableau 8.4 et Figure 8.3), avec les ordinateurs digitaux généralistes placés correctement (au sommet de la hiérarchie des calculateurs), tandis que d’autres objets pouvant être considérés comme (non) calculants peuvent se retrouver en bas ou proche du bas. Il est clair que le seuil se trouve au niveau des calculateurs dédiés spécifiquement, étant donné qu’on peut désirer inclure dans la définition de la computation, des entités qui calculent de simples fonctions comme – seulement – la fonction de succession, ou la somme de 2 entiers, tout en retirant toute capacité de calcul à un “calculateur” uniquement capable de “calculer” la fonction identité.

Cerveaux et calculateurs digitaux (Turing-universels) peuvent tous deux varier considérablement lorsqu’ils sont soumis à deux entrées aléatoires différentes, et il est même possible que cela arrive pour deux entrées arbitrairement proches (selon la même distance de sensibilité des entrées), mais pour des systèmes à faible C , les choses sont différentes. Par exemple, une lampe n’a que deux entrées “aléatoires” possibles : ON et OFF, et le même nombre d’entrées. Pour des systèmes de type un-seul-bloc (incluant les systèmes un-seul-bloc eux-mêmes) les blocs se ressemblent quelles que soient les entrées qu’on puisse imaginer leur fournir. La figure 8.4 montre un comportement de type bloc pour un automate cellulaire élémentaire. Selon Piccinini, ce qui distingue les ordinateurs de la plupart des autres choses, c’est qu’au fond, les ordinateurs sont plus polyvalents que les autres mécaniques calculantes. Il attribue donc une mesure de polyvalence positive au concept de compu-

Objets	Valeur de C	Présence du seuil ($C > \delta$?)
Calculateur digital généraliste électronique (ordinateur)	$C \gg \delta > 0$	Oui
Cerveaux humains	$C \gg \delta > 0$	Oui
Calculateurs à usage spécifique (par ex. calculatrices, machines à énumération successive, etc.)	$C \geq \delta > 0$	Oui/Non
Lampes	$C \sim 0 < \delta$	Non
Rocher	$C = 0 < \delta$	Non

TABLE 8.2 – Une hiérarchie grossière des computations, selon la première approximation de la computation basée sur le coefficient C , avec un seuil réglable δ , déterminant qu'on a affaire à un calculateur si $C > \delta$, sinon ce n'en est pas un. Le symbole " $\gg$ " vaut pour des systèmes pour lesquels (en supposant qu'ils opèrent habituellement de cette façon, par ex. un cerveau humain en pleine possession de ses moyens) on ne peut pas se tromper quant à leur capacités computationnelles en se basant sur leur degré de programmabilité approximée par C . Cela revient à dire que leur valeur C_t^n est strictement plus grande que δ , pour tout δ , pour tout t et n , pourvu qu'on les ait fait tourner assez longtemps (c-à-d assez longtemps pour être plus grand que δ).

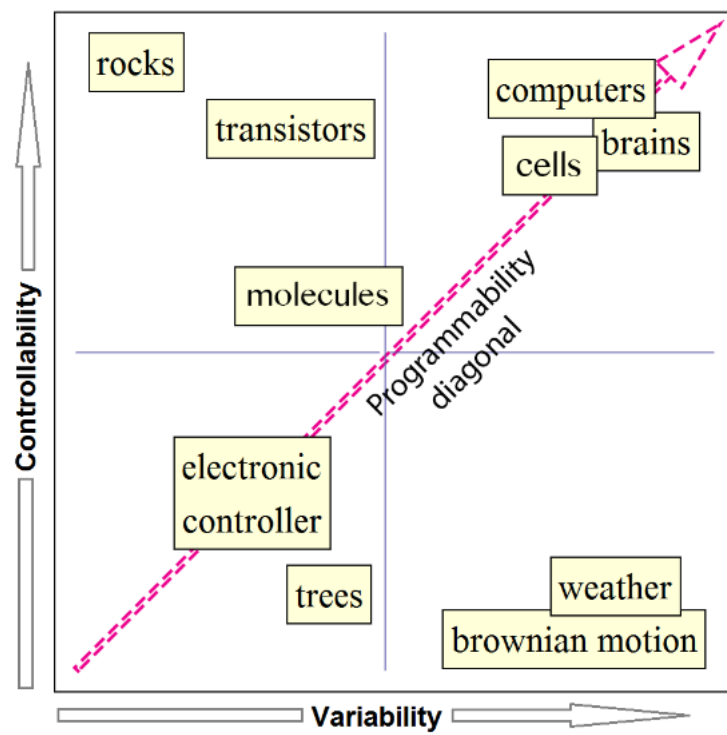


FIGURE 8.3 – Hiérarchie simplifié de programmabilité.

tation (ou d'ordinateur). “Les ordinateurs peuvent faire de l'arithmétique, mais aussi du graphisme, du traitement de texte, des recherches sur internet, et une myriade d'autres choses” dit Piccinini. Et il ajoute : “La versatilité des ordinateurs demande une explication” [Piccinini, 2008]. Certains objets, comme une règle à calcul, possèdent une partie qu'il faut bouger à la main. On peut les appeler des aides à la computation, comme le fait Piccinini. Bien sûr, les règles à calcul doivent avoir des valeurs de C très faibles, sinon nulles, sans l'intervention humaine, et on peut donc les étiqueter comme non-calculateurs, mêmes pour de petits seuils δ .

Ce compte-rendu rend justice aux calculateurs digitaux et aux pratiques des chercheurs en informatique, ainsi qu'au théorème de la calculabilité. D'un côté, les ordinateurs digitaux, les calculatrices, les machines de Turing universelles et non-universelles, ainsi que les automates à états finis, sont des exemples de computation selon la définition proposée. Ils peuvent être reconnus comme ordinateurs, et les ordinateurs digitaux universels peuvent être placés au sommet de la hiérarchie des systèmes computationnels. D'un autre côté, la définition place aussi le concept de programmabilité au centre des pratiques de l'informatique, mais grâce à la complexité algorithmique, on peut aussi définir des classes supérieures de computation basées sur les degrés de Turing, étant donné que les machines abstraites pouvant résoudre le problème de l'arrêt réalisent comportementalement une computation qui ne peut pas être supportée par une machine de Turing qui ne va pas s'arrêter. C'est à dire que la machine oracle s'arrête, mais elle ne va pas s'arrêter pour toutes les computations possibles, elle possède son propre nouveau problème de l'arrêt, d'un degré supérieur, et ainsi de suite, d'où la construction d'une hiérarchie arithmétique sans avoir besoin de descriptions explicites d'états ou de fonctions.

Il est clair que les ordinateurs ne sont pas les seuls mécanismes programmables. Les cerveaux le sont aussi, de même que de nombreux autres systèmes naturels que nous pouvons maintenant contrôler et diriger pour réaliser certaines tâches qu'ils n'étaient pas supposés capables de réaliser naturellement (par ex. par une ingénierie génétique). Un ordinateur est un système qui peut être modifié pour calculer de différentes façons. Je pense qu'une des caractéristiques, peut-être la plus importante, des cerveaux et des ordinateurs, est qu'ils puissent se reprogrammer (de façons différentes mais analogues). Dans l'axe des conclusions de Fodor [Fodor, 1975], mais sans avoir besoin de représentations d'états, si le cerveau est un système programmable, alors c'est un ordinateur sous cette approche comportementaliste. Ceci est paradoxal, car selon la formule de Fodor “pas de computation sans représentation”, selon laquelle, pour qu'on attribue à un système le statut de computationnel,

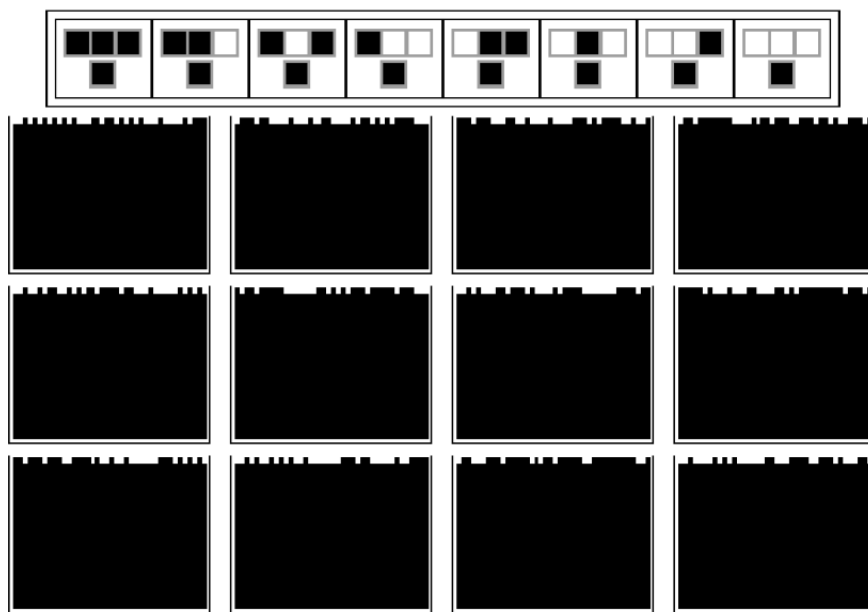


FIGURE 8.4 – Exemples de comportements de type “rocher” par un automate cellulaire élémentaire [Wolfram, 2002] doté de la Règle 256, avec plusieurs configurations initiales “aléatoires”, et évoluant du haut vers le bas. Les évolutions sont précédées d’une icône (au-dessus) illustrant la règle que le système suit pour toute configuration cellulaire possible.

il faut qu’il soit construit comme représentant une information d’une façon ou d’une autre. Selon la vision proposée ici, nous ne devrions ni accepter ni refuser cette formule, car on peut assigner une valeur computationnelle à l’apparition d’un processus naturel, si et seulement si on peut le programmer, quel que soit ce qu’il représente. Comme souligné par Dresner [Dresner, 2010], une représentation de mesure théorique est habituellement accompagnée d’un théorème d’unicité, qui établit comment tous les homomorphismes, en partant de la structure empirique donnée jusqu’à sa version numérique, sont liés entre eux (c’est-à-dire pouvant être obtenus l’un à partir de l’autre). Je vais fournir certaines pistes sur la manière de réaliser cela, dans la réponse aux objections possibles.

Au-delà des formalismes, ce présent compte-rendu sur les ordinateurs et la computation, sert à formuler une taxonomie rigoureuse. Selon cette approche comportementale, toutes les machines de Turing calculant une fonction autre que l’identité sont des “calculateurs” (computers). Sont englobés les esprits et les ordinateurs, tandis que presque tout le reste est exclu, investissant les esprits et les ordinateurs d’un statut spécial. On peut imaginer certains contre-exemples possibles. Pensez au modèle de calcul de la boule de billard. Il est conçu pour fonctionner comme un ordinateur et peut donc être calibré trivialement dans les états d’un ordinateur digital. C’est donc un contre-exemple sémantique de ce que ce compte-rendu est censé faire, c-à-d séparer esprits et ordinateurs (supposés capables de computation), de choses comme les boules de billard, les tableaux et les blocs (supposés incapables de computation). Le modèle de calcul de la boule de billard, en tant que système, est cependant identifié comme computationnel dans son approche comportementale, sans plus de cérémonie.

8.5 Examen des objections possibles

Bien qu’ayant évité le représentationnalisme, ce qui est un avantage de cette approche, nous trouvons que certaines objections au test de Turing, dont celles examinées par Turing lui-même, peuvent servir d’objections à l’approche comportementale de la computation, et que de possibles objections à cette approche comportementale de la computation peuvent aussi servir d’objections au test de Turing. Néanmoins, nous prétendons que cette approche comportementale peut fournir des outils utiles en faveur de la computation naturelle, et nous allons l’utiliser comme base pour un ensemble de mesures en vue de saisir différentes propriétés du comportement dynamique des systèmes naturels, mesures tirées de concepts venant de la théorie al-

algorithmique de l'information et venant de la compressibilité. Ces objections ne sont pas examinées en profondeur ici, car chacune pourrait donner lieu à un article pour elle seule, mais je vais esquisser certaines réponses possibles comme pistes à explorer.

8.5.1 Objections techniques

On va traiter d'abord certaines objections techniques possibles avant de passer aux objections philosophiques. Ces objections et d'autres méritent d'être examinées avec soin, mais il n'est pas question des les traiter toutes ici en profondeur.

La supposition que la compressibilité puisse saisir différents comportements

Une supposition que fait cette première approche d'une définition de la programmabilité, est que les algorithmes de compression sont capables de faire la différence entre différents comportements. D'après la définition proposée, nous obtenons des différences dans les longueurs compressées de l'évolution d'un système. Mais on n'est pas sûr que la longueur de la version compressée d'un système avec une configuration initiale donnée, puisse être différente de la longueur de la version compressée de l'évolution d'un système pour une configuration initiale différente fournissant un comportement apparemment différent. Le problème peut se décrire ainsi : Imaginons qu'on ait deux processus très différents générant des fichiers de données différents, mais que les longueurs de leurs versions compressées utilisant, par exemple, gzip, soient les mêmes. Notre approche aurait l'air de suggérer que deux processus sont les mêmes au niveau comportement, même quand ils pourraient être en fait complètement différents, n'ayant en commun que les longueurs compressées de leurs résultats respectifs, par coïncidence.

Cependant, $K(s)$ est demi-calculable par sa limite haute [Li and Vitányi, 2008]; il existe une série d'algorithmes de compression sans perte approximant $K(s)$: $C_1(s) \geq C_2(s) \geq C_3(s) \dots \geq K(s)$. Ce qui veut dire qu'on peut trouver une série d'algorithmes de compression qui approchent K asymptotiquement. $K(s)$ ne peut pas être plus grand que la version la plus compressée de s . Le théorème d'invariance dans la théorie algorithmique de l'information, garantit que les résultats peuvent se distinguer les uns des autres à la limite, peu importe leurs proximités, grâce à un algorithme de compression approchant K , et jusqu'à un degré

associé de précision (qui peut être large mais l'augmentation de t finit par le dépasser). Plus formellement, le théorème d'invariance établit que si $C_U(s)$ et $C_{U'}(s)$ sont les plus courts programmes générant s en utilisant les machines de Turing universelles U et U' respectivement, leur différence va être associée à une constante additive indépendante de s .

Il est facile de voir que le concept sous-jacent est que, puisque U et U' sont des machines de Turing universelles, on peut toujours écrire un traducteur général (un compilateur) entre U et U' , de telle sorte qu'on puisse lancer l'une ou l'autre machine de Turing et obtenir l'une ou l'autre valeur de complexité, en additionnant simplement la longueur constante du traducteur au résultat.

Cela signifie que finalement, si deux processus sont essentiellement différents quant à leur complexité algorithmique, ils vont avoir des valeurs de C différentes à partir d'un certain temps t , quand on s'approche de K . La mise en garde qu'un système puisse se caractériser de façon imprécise est toujours là, mais le théorème d'invariance garantit que l'approche fonctionne théoriquement, même si en pratique il peut y avoir parfois des erreurs, du fait qu'on se serve d'algorithmes de compression pouvant ne pas "voir" des régularités dans un fichier (par ex. un fichier contenant les décimales de π). Il faut noter que deux évolutions différentes produites par la même règle, comme la règle d'un automate cellulaire, peuvent ne pas avoir nécessairement la même complexité de Kolmogorov (en fait il est peu probable qu'ils aient une même complexité si les évolutions paraissent différentes), car le système ou la règle est $S(i)$, et non pas S tout seul. $S(i)$ étant S pour la configuration initiale i (par ex. l'automate cellulaire élémentaire de Règle 30 [Wolfram, 2002] partant d'une cellule noire est un autre système que la Règle 30 partant d'une répétition de dix fois 01). D'après l'universalité de Turing, on sait que $S(i)$ peut toujours se réécrire en S' , qui est un système à entrée vide qui se comporte comme S avec l'entrée i , où il est clair que $S \neq S'$, et cette différence est finalement capturée par la différence entre $K(S)$ et $K(S')$, qui sont les longueurs des plus courts programmes produisant S et S' .

Le choix de l'énumération des configurations initiales

L'interrogateur est une part importante de cette approche basée sur Turing, c'est pourquoi les configurations initiales d'entrée sont essentielles – leur rôle est analogue à celui de l'interrogateur questionnant le système. En général, on peut toujours altérer une énumération E pour faire en sorte qu'un système se comporte d'une certaine façon pendant un temps limité, car on peut toujours lancer un système puis choisir des conditions initiales pour lesquelles le système va agir d'une certaine manière en concevant une autre

énumération E' pour laquelle les premiers membres E'_t soient membres de E , mais triés selon $t = 1, \dots, n$ de sorte que le système agisse de façon voulue pour les premiers n éléments. Ainsi, quelle est la sensibilité au choix de l'énumération initiale d'entrée, pour cette approche inspirée du test de Turing dans le cas du problème de la computation naturelle ? On peut rendre n aussi grand qu'on veut, mais le comportement limite d'un système va toujours se trouver au-delà de n . Est-ce que cela suffit pour garantir qu'à un certain moment (par ex. n), le système va commencer à se comporter de façon "naturelle" ? Imaginons qu'on sache qu'un système se comporte d'une certaine façon pour des configurations initiales de longueur paire. On pourrait alors concevoir un E , tel que toutes les configurations initiales soient de longueur paire. Mais E doit atteindre toute configuration initiale possible en un temps fini, donc il n'y a pas moyen de concevoir un E de façon à ce qu'il parcoure toutes les entrées de longueurs paires, puis toutes les entrées de longueurs impaires, en un temps fini. Il n'y a pas moyen de duper/contourner l'analyse limite du comportement d'un système en altérant ses configurations initiales sur plus qu'un nombre fini d'entrées.

La question générale de l'énumération appropriée des entrées pour un système mérite d'être étudiée, tout particulièrement pour les systèmes naturels, étant donné qu'il n'est pas toujours évident de savoir ce que peut être une énumération d'entrées pour un système naturel (on en vient par exemple à se demander s'il faut discrétiser des paramètres en valeur continue, pour qu'un algorithme de compression puisse les analyser). Un problème évident est celui des "systèmes encryptés." Qu'en est-il si un système calculant programmable efficacement se présente intentionnellement sous un jour aléatoire et inefficace ? Par exemple un système Turing-universel (disons la Règle 110 [Wolfram, 2002, Cook, 2004]) se comportant comme un autre système à l'air aléatoire (tel que la Règle 30 appartenant au même espace de règles). C'est toujours la Règle 110, mais la question est de savoir si on serait capable de l'identifier et de programmer cette Règle 110 si elle se comporte comme la Règle 30. Il se peut qu'on ne soit capable de la reconnaître comme Règle 110 uniquement si on connaît la fonction de décryptage, ainsi l'algorithme de compression peut être trompé/dupé. C'est la même chose que de pouvoir passer un "test de stupidité" quand le système est assez malin pour savoir comment avoir l'air stupide, ou alors il est réellement stupide (on ne peut cependant pas passer un test d'intelligence ou de computation sans être intelligent ou capable de calculer). Il est important de traiter la question des "systèmes encryptés" apparaissant dans la nature. Ceci est certainement lié à une caractéristique que, d'après moi, devrait posséder cette approche comportementale, à savoir une subjectivité dépendante de l'observateur, et

lié aussi à la question de l'énumération des conditions initiales, ainsi qu'à la question d'une certaine sorte de besoin minimal de représentation.

8.5.2 Objections sur les fondements

Il est intéressant de voir comment certaines objections servent immédiatement d'arguments contre l'approche de Turing sur l'intelligence des machines et contre notre approche sur la computation naturelle, alors que d'autres non (par ex. l'Objection Mathématique de Searle [Searle, 1990] et Penrose [Penrose, 1999] qui d'évidence ne semble pas s'appliquer à la question de la computation). D'autres exemples concernent les arguments théologiques et les arguments sur la conscience, qui travaillent tous deux contre l'intelligence des machines et contre la computation naturelle, en faisant endosser aux humains et aux choses naturelles une *qualia* (une qualité sensible irréductible), la dite *qualia* supposée ne pas pouvoir être concomitante avec le domaine de la computation digitale. Toutefois, ces objections fonctionnent différemment, car dans le cas de l'intelligence des machines, elles veulent "sauvegarder" l'essence de l'être humain, lui attribuant des qualités irréproductibles comme la conscience. Tandis que dans le cas de la computation naturelle, elles fonctionnent pour "sauvegarder" l'essence, la nature de la computation digitale. L'avantage de mon approche, comme celle de Turing, c'est qu'il y a moins de gens décidés à défendre les machines que de gens décidés à défendre les humains, bien qu'on débattent chaudement dans les deux directions. Cependant, la tendance courante en computation est une plus grande ouverture à la possibilité que des objets et des systèmes autres que des ordinateurs électroniques, réussissent à computer.

Une certaine représentation est nécessaire

Il est intéressant de noter qu'on a besoin d'une certaine représentation des résultats d'un système, avant d'alimenter l'algorithme de compression (voir figure 8.5). Qu'en est-il de l'introduction ou de la simplification de complexité dans le processus d'encodage partant du langage du système vers le langage d'un ordinateur digital, en implémentant l'algorithme de compression sans perte? C'est en effet le cas, et cela implique qu'il existe une certaine communication et une certaine projection entre le système naturel et l'ordinateur digital, en implémentant l'algorithme de compression sans perte, mais cette projection est d'une nature très différente des projections d'états et de fonctions entre les systèmes eux-mêmes. Cette représentation est-elle toujours possible? D'un côté, on peut toujours discrétiser des données. De

l'autre côté, on sait qu'un langage discret peut toujours se traduire en binaire. Donc, dans un sens technique, c'est toujours possible. Ceci est lié à la discussion précédente sur la question de savoir si un système universel peut émuler ou faire semblant d'être un système apparemment aléatoire, pour cacher ses capacités de programmabilité, et ce que cela voudrait dire.

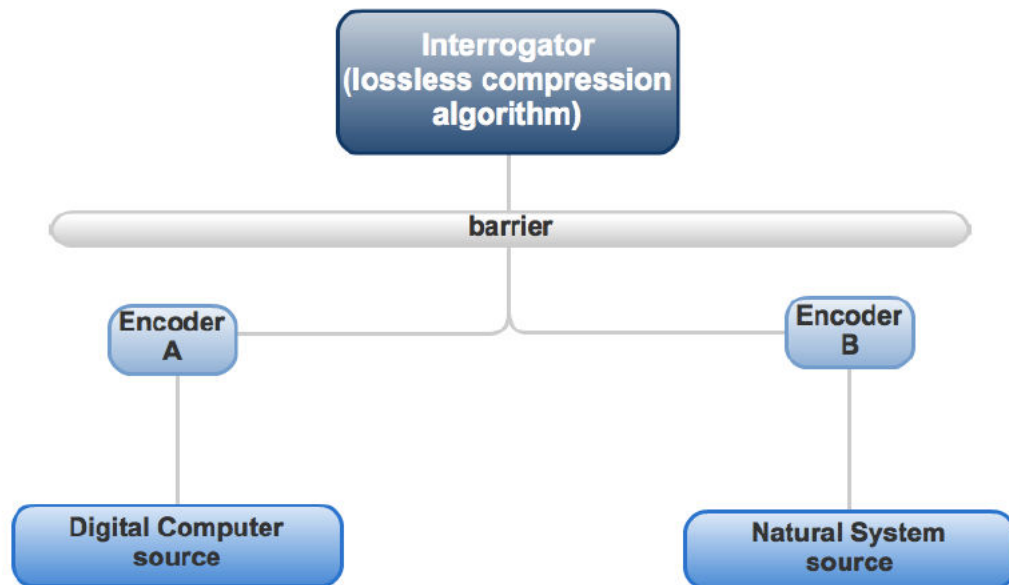


FIGURE 8.5 – Quelle est la nature des encodeurs ? Ils travaillent dans les deux directions, encodant d’abord les “questions” propres à chaque système, pour ensuite alimenter l’algorithme de compression sans perte dans le bon format. Les encodeurs A et B peuvent être de nature très différente. Il semble toujours possible d’avoir affaire à des encodeurs simples, mais les questions sur leur implémentation et leur rôle restent légitimes

Toutefois, la proposition avancée ici est différente de ce que demanderait une forme forte de représentationnalisme, où il faudrait connaître les états d’un système pour les mettre en relation, et où on est complètement dépendant d’une complète (et improbable) connaissance des états d’un système naturel. Alors qu’on n’a besoin ici que d’une faible traduction du langage de sortie d’un des systèmes dans l’autre, représentée par les boîtes encodeuses dans la figure 8.5. Il ne faut pas voir les encodeurs comme des inconvénients, on a sans cesse affaire à eux en informatique sous la forme de compilateurs. Dans le développement des systèmes, différents langages de programmation sont utilisés pour différents objectifs, en des lieux différents d’un système.

Même si on pouvait les simuler les uns vers les autres en tant que langages Turing-complets, très peu d'applications non-triviales d'aujourd'hui sont totalement développées dans un seul langage de programmation.

Programmabilité

Une seconde réaction immédiate consiste à se demander si placer la programmabilité au cœur de la définition de la computation n'est pas une supposition trop forte. Car on peut penser à des systèmes naturels et artificiels pouvant avoir l'air non programmables, tout en étant prêt à admettre qu'ils peuvent calculer (par ex. des réseaux discrets de neurones). Je pense que cette objection vient d'un mélange entre la signification standard de la programmation et la signification comportementale que j'avance ici. Alors qu'il est vrai que pour bien des systèmes naturels et artificiels, un concept de programmabilité est difficile à déterminer, le concept de programmabilité avancé dans ce chapitre concerne la possibilité, par tout moyen, de faire se comporter un système d'une manière autre que celle qu'il avait déjà. Dans ce sens, par exemple, un circuit logique ou un ensemble de processus peuvent ne pas être considérés comme computation s'ils sont incapables de réagir à des stimuli externes, ou si l'observateur est incapable de témoigner d'une telle interaction, que ce soit dans la conception ou dans le lancement d'un processus de calcul.

L'homme-machine et les objections d'intentionnalité

Quand Julien Offray de La Mettrie [[de La Mettrie and Thomson, 1996](#)] amenait la méthode de Descartes à ce qu'il prétendait être sa conclusion logique dans son homme-machine, l'argument était que Descartes, en essayant de défendre la théorie d'une âme humaine en reléguant le comportement mécanique aux animaux, agissait en fait contre les humains. Car si les animaux sont capables de se nourrir, de bouger et d'interagir entre eux, il n'y a rien à proprement parler qui empêche de voir le comportement humain comme la conséquence d'un comportement mécanique. Dans le test de Turing, on voit une inversion semblable de l'argument, là où ce n'est pas l'intelligence de la machine qui est testée, mais l'intelligence de l'être humain, non pas parce les questionneurs entretiennent le doute sur l'intelligence des humains puissent ne pas être intelligents, mais parce que les mécanismes qui commandent l'intelligence humaine peuvent s'avérer du même ordre que ceux qui commandent les ordinateurs d'aujourd'hui.

Searle pose le problème du sens interne ou “intentionnalité” [Searle, 1990]. Harnad [Harnad, 1990] le définit comme le problème de base des symboles (*grounding symbol problem*) ou comment les symboles prennent signification. Je considère cette objection comme faible dans le contexte qui nous occupe (bien que largement inexplorée), car si on l’accepte, la définition du calcul perd tout son sens dans le contexte physique (on sait qu’on peut programmer certaines choses naturelles, ces choses pourraient être considérées comme des ordinateurs quand elles calculent pour nous, et pas du tout quand ils ne le font pas, alors qu’elles restent physiquement les mêmes). Dans le test de Turing, une certaine “intention” est désirable, car Turing essaye de convaincre son lecteur qu’il y a pas d’argument de principe pour qu’une machine échoue à un test d’intelligence si on améliore progressivement ses performances quant à imiter le comportement humain intelligent. Il est clair aussi que les ordinateurs électroniques sont assemblés dans le but de computer, donc aucun n’effort n’est fait pour supporter de l’intentionnalité.

Dennett a suggéré [Dennett, 1981] qu’il semblerait qu’une représentation explicite ne soit pas nécessaire pour expliquer des attitudes propositionnelles. Par exemple, pendant une partie d’échecs contre un logiciel informatique, des attitudes du genre “Je pense que je devrais déplacer la reine sur la gauche” sont souvent attribuées à l’ordinateur. Pourtant personne ne songerait à prétendre que l’ordinateur pense ou croit réellement, comme nous le faisons, et il est clair d’après moi que notre approche comportementale de la computation est compatible avec ce point de vue, tout en étant neutre sur les questions d’intentionnalité, car on ne s’intéresse qu’aux façons dont un système semble se comporter, et non pas s’il le fait “réellement” ainsi (ce qui veut dire qu’il a l’intention de faire ainsi, peu importe si on a affaire à des ordinateurs ou des systèmes naturels, cerveaux inclus).

L’objection orientée-observateur

Une réaction immédiate, ainsi qu’une objection possible à notre approche, concerne l’applicabilité d’une telle définition comportementale (orientée-observateur), étant donné la possibilité arbitraire du choix de δ (voir par ex. le Tableau 8.4). Selon certains arguments, la computation est relative-à-l’observateur, que ce soit au sens où de nombreux systèmes physiques implémentent de nombreuses computations (Putnam [Putnam, 1988]), ou au sens où presque tous les systèmes physiques implémentent toutes les computations (Searle [Searle, 1992]). On peut voir par exemple certains objets physiques capables d’implémenter toute computation de n’importe quelle complexité. Ainsi les parois du [Searle, 1992] de Searle, qui implémentent

son programme de traitement de texte. Puisque la description physique d'un objet sous-détermine sa description computationnelle de cette manière, on juge la computation comme relative-à-l'observateur [Searle, 1992].

C'est bien sûr une objection légitime, qui s'applique aussi aux autres approches comportementalistes pour d'autres notions, telle que la notion d'intelligence, et au test de Turing. Cependant, j'ai suggéré (section 2.1) qu'une mesure du comportement limite était possible, et que même si δ est très large, on peut toujours s'en sortir et le dépasser avec le temps pour des systèmes qui soient indubitablement des appareillages capables de calculer selon l'approche par la programmabilité (ordinateurs et cerveaux), et toujours s'arranger pour que des appareillages triviaux comme les lampes et les rocs échouent à entrer dans la catégorie ordinateurs, laissant un espace intermédiaire flexible pour des systèmes pouvant ou non être considérés, subjectivement, comme ordinateurs.

On considère habituellement que l'un des inconvénients de la complexité de Kolmogorov, est sa dépendance par rapport aux langages de programmation ou par rapport aux machines de Turing. Dans notre approche, on tire un avantage concret de cette propriété de la complexité de Kolmogorov, car elle est assortie d'une approche comportementale de la computation garantie non relative-à-l'observateur (ou à-la machine). Ceci parce que la complexité de Kolmogorov (basée sur la taille des programmes ou sur l'algorithmie) n'a de sens qu'une fois fixé la machine de Turing ou le langage Turing-complet. D'un autre côté, comme la complexité (K) de Kolmogorov est incalculable (un autre inconvénient communément identifié), ou plus précisément, demi-calculable par le haut, c'est ce que l'algorithme de compression "observe" qui approxime K , ce qui tourne à notre avantage en saisissant le comportement qualitatif d'un système computationnel pour le mesurer quantitativement.

Piccinini défend l'idée que toute définition raisonnable de la computation doit être objective. Je ne pense pas, cependant, que cela doit être une condition sine qua none d'un compte-rendu raisonnable sur la computation, ni qu'échouer à atteindre ce critère objectif rende un tel compte-rendu vide ou trivial. En fait, je pense que la computation est intrinsèquement orientée utilisateur/observateur, à la fois en pratique et en théorie. En pratique, la computation concerne principalement, sinon entièrement, la programmation de systèmes. D'un côté, programmer des systèmes est intentionnel (motivé par le désir de faire qu'un ordinateur se comporte d'une façon particulière), même si l'intentionnalité n'est pas essentielle à la computation. D'un autre côté, la théorie impose de la subjectivité de différentes façons. Le problème de l'arrêt peut se lire comme une propriété relative-à-l'observateur pour les systèmes computationnels, étant donné qu'on ne peut généralement pas sa-

voir si une computation va s'arrêter, sauf à dérouler le système pendant un nombre spécifié d'étapes – chose qui dépend de la volonté de patience de l'observateur, si le système ne s'arrête pas avant ce nombre spécifié d'étapes. Le problème n'est pas réservé à l'arrêt, mais s'étend à l'accessibilité en général, ce qui en fait la question de savoir si un système va atteindre une certaine configuration. La computation universelle est subjective au sens où on doit décider de stopper une computation et s'arranger avec le fait qu'on ne puisse jamais savoir si une telle computation va jamais s'arrêter ou bien va atteindre une certaine configuration.

Le problème de l'arrêt impose de la subjectivité à tous les niveaux

Le problème de l'arrêt est le problème de décider si une computation va s'arrêter ou non. Le problème de l'arrêt implique que les computations peuvent se diviser en 2 catégories : réductibles et irréductibles. D'une part les computations suffisamment simples pour pouvoir déterminer si elles vont s'arrêter ou ne jamais le faire, et d'autre part les computations pour lesquelles la seule option est de les lancer en attendant qu'elles s'arrêtent seules, ce qui peut évidemment prendre un temps infini. Les computations irréductibles peuvent aussi être classées en 2 types : les computations qui ne s'arrêtent jamais et les lancer ne sera donc d'aucune aide, et les computations qui s'arrêtent au bout d'un temps t , mais pas moyen de connaître t sinon en faisant tourner la computation pendant au moins t . Cette caractérisation donne clairement un rôle important à l'observateur, du fait qu'il existe des computations pour lesquelles on ne peut savoir si elles s'arrêtent uniquement en les faisant tourner, introduisant une composante subjective, en l'occurrence le fait que l'observateur doive décider du temps qu'il est prêt à attendre avant de couper court à son observation de l'activité de la computation, en faisant ainsi une supposition d'information sur la caractéristique d'arrêt (ou de non-arrêt) de la computation. On peut voir maintenant l'importance d'un observateur dans le compte-rendu d'une computation, même dans le cas le plus classique d'insolubilité du problème de l'arrêt. C'est encore plus évident en considérant d'autres phénomènes, tel que l'accessibilité, à savoir si une computation va atteindre une certaine configuration, où pour certaines computations seul un observateur voulant lancer et observer la computation peut répondre.

L'indécidabilité du problème de l'arrêt affecte toutes les notions théoriques et pratiques liées à la computation. Par exemple dans la complexité de Kolmogorov, on ne peut jamais dire si un objet est aléatoire (on peut dire si un objet est simple s'il a été compressé, mais pas l'inverse). Cela ne rend pas

la complexité algorithmique inutile. En fait, c'est la mesure de cette propriété relative-à-l'observateur (relative aux algorithmes de compression qui peuvent ou non "voir" des régularités dans les données) qui est la plus utile – pour toutes sortes d'applications, incluant la classification des espèces animales et des langages par compressibilité, la détection des séquences génétiques, la détection de la fraude et du plagiat. Dans la complexité finie de Kolmogorov, l'aléatoire fini est dans l'œil de l'observateur, au sens où toute séquence finie peut toujours faire partie d'une chaîne aléatoire ou non-aléatoire. D'où la qualité d'être aléatoire est observateur-dépendante, tout comme dans le cas du problème de l'arrêt.

Je pense que c'est le déni du rôle de l'observateur qui rend vide ou triviale la notion intuitive de computation. L'observateur joue un rôle essentiel dans la définition de la computation. Notre définition de la computation rend ceci explicite, dans l'intention de généraliser et caractériser la computation naturelle.

Selon cette approche, la computation est relative-à-l'observateur (en accord avec de nombreux auteurs approuvant le computationnalisme), tout comme l'intelligence est relative-à-l'observateur dans le cas du test de Turing. Nous trouvons que certaines objections au test de Turing, dont celles relevées par Turing lui-même, peuvent servir d'objections à cette approche comportementale de la computation (nous allons traiter certaines d'entre elles), et que des objections possibles à l'approche comportementale de la computation peuvent aussi servir comme objections au test de Turing. Néanmoins, nous prétendons que l'approche comportementale de la computation peut fournir des outils utiles pour la computation naturelle, et nous allons les utiliser comme base pour fabriquer une panoplie d'outils de mesures quantitatives – basée sur des concepts tirés de la théorie de l'information algorithmique et de la compressibilité – en saisissant différentes propriétés qualitatives des systèmes naturels.

Paradoxalement, l'approche comportementale n'explique pas le comportement d'un système, du moins pas entièrement, car on peut expliquer une partie du comportement d'un système dès qu'une première analyse comportementale est réalisée, mais pas de la manière à laquelle on pourrait s'attendre en suivant Smith ou Piccinini, car cette approche n'a pas l'intention d'être une théorie du calcul/computing, et elle n'a pas été montée pour pleinement rendre compte des causes du comportement d'un système, mais seulement rendre compte de son comportement apparent. On a démontré que l'approche est applicable et peut fournir des indices au sujet des propriétés de systèmes dynamiques, dont les états internes sont inaccessibles, et dont on n'a aucune idée des liens pouvant exister entre système naturel et système computation-

nel abstrait. Et cela marche tout aussi bien pour des systèmes qu'on connaît et dont on peut étudier les états internes en détail, en produisant toutes sortes de liens vers d'autres modèles de computation, comme nous l'avons montré en utilisant les automates cellulaires et par la façon dont des mesures basées sur cette approche comportementale nous ont permis de caractériser des transitions de phase ou des vitesses de transfert d'information, à partir d'une perspective purement comportementale.

Prenons l'exemple d'avoir à calculer les exposants de Lyapunov d'un système naturel. Même si le système peut se décrire comme un système dynamique pour lequel les orbites sont descriptibles, cela suppose déjà qu'on soit capable de représenter une telle dynamique. Bien sûr l'approche comportementale suppose aussi qu'on puisse saisir le comportement du dit système, mais elle ne suppose pas la pleine connaissance de l'évolution précise du système. En fait, on peut aller jusqu'à analyser un système à un moment donné sans avoir à remonter des temps intermédiaires (cela va bien sûr impacter le résultat final, car l'amélioration de ce résultat est directement proportionnelle au temps d'observation du système).

Si l'observateur est essentiel pour la définition de la computation, on doit reconnaître que la question la plus générale consistant à se demander si l'univers compute, n'a pas de sens. Car aucune définition de l'univers ne permet les stimuli extérieurs (externes à l'univers), ni ne permet que les résultats de l'univers se situent extérieurement à lui, pour qu'un observateur puisse les évaluer.

L'implémentation a-t-elle de l'importance ?

La question de l'implémentation d'une computation semble ne pas avoir été sérieusement prise en compte jusqu'à ce que la critique du computationnalisme mette en avant certains arguments pour attirer l'attention sur le fait que de très nombreux systèmes physiques implémentent de nombreuses computations, sinon toutes. De tels arguments ont été présentés par Putnam [[Putnam, 1988](#)] et Searle [[Searle, 1992](#)]. Selon Putnam (et son éponyme Théorème de Réalisation de Putnam), "pour tout système S ordinaire et ouvert, pour tout automate M à état fini (sans entrée/input et sans résultat/output), pour tout nombre n d'étapes computationnelles de l'automate M , et pour tout intervalle I de temps réel (divisible en n sous-intervalles), S réalise n étapes computationnelles de M pendant I ." Et selon Searle (ce qu'il est coutume d'appeler parfois la Thèse de Searle), "pour tout programme et pour tout objet (physique) suffisamment complexe, il existe une certaine

description de l’objet selon laquelle l’objet en question est en train d’implémenter le programme.”

Dans l’axe de la question posée par Chalmers [Chalmers, 1996], qu’est-ce qui fait qu’un rocher compute quelque chose (ou rien du tout) plutôt que tout ? Il semblerait, au premier abord, que ce que la computabilité (abstraite) et la computation (concrète) aient en commun, ce soit une sorte de description logique. Mais seule la caractérisation de la computation (concrète) n’est pas épuisée par une description purement logique, donc l’implémentation a de l’importance. Et elle a de l’importance dans mon approche, étant donné que le rocher pourrait être capable de faire n’importe quelle computation (pensez à l’utilisation de ses particules pour construire un dispositif plus programmable). Mais il ne va pas le faire au niveau auquel on doit le décrire comme rocher, par contre si on le regarde à travers une loupe inspirée du test de Turing, et qu’on essaye de le faire se comporter d’une façon ou d’une autre, cela revient à le programmer pour qu’il se comporte différemment pour différents stimuli extérieurs (voir la figure 8.4 pour le comportement type “bloc” d’un système abstrait.)

Les lois n’ont pas de caractère distinct

On a suggéré [Parsons, 2012] que j’avais assigné un statut spécial aux lois physiques, et notamment aux programmes informatiques. C’est une objection compréhensible, mais en fait elle vient d’une idée fausse de ma position. Le quiproquo consiste à croire qu’en reliant des lois à des programmes informatiques et non à des données, je donne aux lois un statut spécial, immortel et inchangé. Mais on peut écrire les programmes informatiques en bits. Et comme je l’ai expliqué dans la section 8.2, Turing a prouvé que les programmes informatiques et les données ne sont pas essentiellement différents ; on peut toujours les interchanger. Ce qui fait qu’il est possible d’écrire le tableau de transition d’une machine de Turing sous la forme d’une entrée pour une machine de Turing universelle, ou de construire un tableau de transition (une machine de Turing) à partir de la description d’un programme informatique.

En probabilité algorithmique, il n’y a qu’une seule supposition forte au sujet de la distribution des objets. Ce que la distribution universelle de Levin est supposée indiquer est la probabilité d’une chaîne ayant été générée par un programme, mais on doit aussi faire cette supposition pour la distribution des programmes, afin de pouvoir parler du choix d’un programme aléatoire. Et c’est la seule supposition possible non-informée – la distribution uniforme. Ce qui fait que tous les programmes de même longueur ont la même probabilité

d'apparaître en tant que produits du hasard. Mais à part cela, on est libre d'interchanger les programmes. Les lois physiques n'ont rien de spécial. On peut les voir comme soulignant ou résumant une régularité dans les données (le monde), et les données peuvent changer, donc les lois physiques aussi.

La question de l'échelle

Dans le monde réel, les choses sont constituées d'éléments plus petits, à moins que ce soient des particules élémentaires. On doit donc étudier le comportement d'un système à une échelle donnée, et non pour toutes les échelles possibles, autrement la question devient vide de sens. Car les éléments d'un objet physique sont des molécules, puis finalement des atomes et des particules qui possèdent leur propre comportement, et sur lesquelles on peut aussi poser la question de la computation. Cela signifie qu'un C -ordinateur peut avoir un C bas ou nul à une certaine échelle, et contenir des C' -ordinateurs avec $C' > C$ à une autre échelle (pour laquelle l'objet original n'est plus le même dans son ensemble). On voit couramment des organisations concrètes de dispositifs computationnels divers avec $C' \leq C$ à une certaine échelle. Par exemple un ordinateur est constitué d'éléments plus simples, chacun d'eux, à un certain niveau macroscopique mais indépendamment de l'ordinateur interconnecté, est d'une richesse comportementale moins élevée et peut être qualifié d'un C moins élevé. Autrement dit, la définition n'est pas additive, au sens où un C -ordinateur peut contenir ou être contenu dans un autre C' -ordinateur avec $C' \neq C$. Dans le monde physique, selon cette approche qualitative, les choses peuvent calculer ou non, en fonction de l'échelle à laquelle on les étudie. Dire qu'un tableau compute n'a de sens qu'à l'échelle du tableau, et en tant que C -ordinateur il aurait un C très limité, ce qui veut dire un éventail très limité de comportements, car on peut à peine le programmer pour faire quelque chose d'autre.

La définition comportementale n'est pas insensible à l'échelle. Quelque chose peut ou ne peut pas calculer à un certain niveau de description, mais il se peut qu'il calcule à un autre niveau plus macro- ou plus microscopique de description. Aussi, le concept d'objet n'est pas échelle-invariant (nous appelons les choses par des noms différents quand on change d'échelle, par ex. nous appelons atomes les constituants d'un rocher, et l'agrégation de H_2O dans un liquide forme l'eau).

L'objection du traitement-par-lots (batch process)

Un traitement-par-lots est l'exécution d'un programme sur un ordinateur sans avoir besoin d'aucune intervention extérieure. Ce type de système devrait passer inaperçu face à l'approche comportementale que nous proposons, étant donné l'insensibilité d'un tel système à tout stimulus extérieur, car il est programmé pour réaliser une tâche sans interagir avec quoi que ce soit avant qu'il ne stoppe en produisant un certain résultat, si résultat il doit y avoir. Pendant ce temps, le processus peut avoir l'air de ne rien faire du tout, mais c'est simplement une apparence, et il existe des moyens pour que l'observateur soit sûr que la computation est en route. Au plus bas niveau, par la consommation et la production de ressources externes, comme l'énergie et la chaleur (qu'on peut manipuler pour changer le comportement du processus, par exemple pour l'arrêter), et à un autre niveau, en contrôlant le processus pendant un temps suffisamment long. L'exemple du traitement-par-lots n'est une objection valide qu'entre le temps $t = 1$ quand le processus est initié (il le faut), et le temps $t = n - 1$, car au moins à un moment $t = 0$ ou $t = n$ (s'il s'arrête et produit un résultat), on attend l'apparition d'une interaction avec l'extérieur. Donc, tandis que la définition comportementale peut échouer à identifier certains ordinateurs, la définition de la limite comportementale semble insensible à cette objection, sauf dans la mesure où, pour toute bonne raison (spécifique), on serait amené à considérer quelque chose qui compute, comme ne calculant pas, car déconnecté du monde extérieur où vit l'observateur.

L'éventualité de la mécanique quantique

En se servant de la probabilité algorithmique (PA), S. Lloyd affirme [Lloyd, 2002] :

Je suggérerais, ici simplement comme une métaphore, mais aussi sur la base d'un programme scientifique visant à explorer la capacité de calcul de l'univers, qu'il existe aussi une explication raisonnable quant à savoir pourquoi l'univers est complexe. Il se fait programmer par de petites fluctuations quantiques aléatoires, du même genre que les fluctuations quantiques qui font que notre galaxie est là plutôt qu'ailleurs.

– (S. Lloyd, 2002)

Nous ne savons pas si la PA peut s'adapter à une version quantique, mais nous savons bien qu'on n'a pas besoin de fluctuations quantiques pour

générer de la structure algorithmique [Delahaye and Zenil, 2012], que Llyod essaye d'expliquer en se basant sur la mécanique quantique.

La supposition forte dans le contexte de la computation classique et de la mécanique classique, c'est le déterminisme. L'effondrement de la fonction d'onde en mécanique quantique et le problème de la mesure peuvent mettre en doute le déterminisme à deux différents niveaux, mais sinon la mécanique classique impose le déterminisme dans l'univers (macroscopique). La mécanique classique (newtonienne) garantit le résultat déterministe (le problème est de générer le même input/entrée). Faire tourner une computation deux fois avec la même entrée génère le même résultat grâce à exactement le même cheminement, tout comme l'aurait fait un système classique suivant les règles de la mécanique classique (ce qui est impossible en pratique à cause de la précision limitée de la mesure des conditions initiales).

Liens avec la complexité computationnelle

A la lumière de cette recherche, on peut maintenant trouver un lien intéressant entre la mesure C et la complexité computationnelle traditionnelle, où on s'occupe des ressources nécessaires pour qu'une computation puisse se faire. C donne des indices pour savoir si un système peut être Turing-universel, mais pas pour savoir s'il peut ne pas l'être, car l'universalité requiert variabilité et sensibilité aux stimuli extérieurs pour programmer une computation. Donc C est grandement influencée mais pas directement liée à l'universalité, étant donné que l'universalité va garantir que $\lim C_t^n = \infty$ pour $t, n \rightarrow \infty$, mais une valeur positive de C ne garantit pas l'universalité mais la sensibilité, qui dans ce contexte est une mesure de la capacité du système à être programmé pour faire différentes computations (même limitées) en transférant de l'information à partir d'une entrée vers une sortie. Et C dépend finalement de la façon dont on calcule C pour un nombre fini de configurations initiales et un nombre fini d'étapes, donc les systèmes qui calculeraient sur un rythme lent pourraient être mal classés pour certains t et n suffisamment petits. On peut aussi cependant voir C comme mesurant l'efficacité d'un système à être programmé. On peut ainsi relativiser ce concept qui introduit des classes temporelles de complexité. On peut donc dire d'un système de valeur C croissant en temps linéaire qu'il est efficace, mais il ne l'est pas s'il grandit en un temps logarithmique.

8.6 Remarques en conclusion

Ce chapitre a examiné le problème de la reconnaissance de la computation. Il répond partiellement à certains des besoins que toute définition de la computation doit satisfaire, selon plusieurs auteurs (par ex. [Scott, 1970, Piccinini, 2008]), mais j’ai souligné que certaines propriétés ne sont pas nécessaires et ne devraient pas être demandées ni attendues, particulièrement dans le nouveau contexte de la computation naturelle et de la biologie artificielle. Les modèles computationnels peuvent être très utiles, même quand on ne connaît pas tous les détails d’un système. Le but de la biologie des systèmes est de comprendre, par exemple, les propriétés fonctionnelles et le comportement des organismes vivants, tandis que le but de la biologie synthétique est de concevoir, contrôler et programmer le comportement de systèmes vivants, même sans connaître les détails des systèmes biologiques en question. Dans la lignée du test de Turing sur l’intelligence, cette approche semble être utile pour explorer, de façon quantitative, des propriétés qualitatives des systèmes calculants, et comme elle place la programmabilité au centre de la computation, elle peut servir de fondement à une théorie de la computation naturelle.

Chapitre 9

L'approche algorithmique de la cognition

Nous passons revue des concepts qui sont à l'avant-garde des recherches reliant la cognition humaine, animale et artificielle, aux processus de computation et d'information – en partant du test de Turing jusqu'à l'argument de la Chambre Chinoise de Searle, en passant par la Théorie Intégrée de l'Information et la complexité computationnelle et algorithmique. Nous commençons par défendre l'idée que passer le test de Turing est un problème computationnel trivial et que sa difficulté pragmatique éclaire la nature computationnelle de l'esprit humain, plutôt qu'elle ne met au défi l'intelligence artificielle. Puis nous examinons les mesures basées sur l'algorithmique et la théorie de l'information, mesures que nous proposons pour quantifier et caractériser la cognition sous différentes formes. Ces mesures sont capables de rendre compte des biais connus du comportement humain, et revendiquent ainsi une vision computationnelle algorithmique de la cognition, suggérée pour la première fois par Turing, mais enracinée cette fois dans le concept de probabilité algorithmique qui se base sur l'universalité computationnelle, tout en étant indépendant du modèle computationnel, avec l'avantage d'être un modèle théorique prédictif et testable du comportement cognitif.

9.1 Le modèle algorithmique de l'esprit

Judea Pearl, un théoricien reconnu de la causalité, pense que tous les scientifiques de l'informatique sont en quelque sorte des psychologues frustrés car ils apprennent à partir d'eux-mêmes (et à partir d'autres) en écrivant des

programmes informatiques qui sont des émulateurs du comportement humain intelligent [Clark, 2012]. Comme Pearl le suggère, les programmes informatiques sont une version avancée d’une composante intellectuelle importante de nous-mêmes et ils réalisent des calculs pour nous, quoique pas toujours de la façon dont nous l’aurions fait. Pearl prétend qu’en nous reproduisant nous-mêmes, nous extériorisons une partie de notre comportement ; nous reflétons notre propre intérieur en lui permettant de devenir objet d’investigation. Nous sommes capables de contrôler les conséquences des modifications de notre esprit, en changeant le code d’un programme informatique qui est le reflet de notre esprit.

Peut-être que le premier de ces psychologues frustrés a été Alan Turing, en essayant d’expliquer le comportement humain au moyen de processus mécaniques, d’abord par sa féconde étude sur la computation universelle [Turing, 1937], mais aussi plus tard dans son article marquant faisant le lien entre computation et intelligence grâce à un jeu d’imitation [Turing, 1950]. D’un autre côté, l’Intelligence Artificielle a cherché à automatiser des aspects considérés comme possiblement intelligents dans le passé, dans l’esprit du premier papier de Turing sur la computation universelle. Cela a conduit l’évolution des programmes informatiques à passer de simples calculateurs arithmétiques à des machines capables de jouer aux échecs à des niveaux supérieurs aux capacités humaines, en leur faisant répondre à des questions presque aussi bien – et dans certains domaines mieux – que les performances humaines, à des machines capables de traduire des langues et de reconnaître des visages mieux que les standards humains. D’un autre côté, le débat philosophique incarné par l’article tardif de Turing sur l’intelligence des hommes et des machines, s’est orienté très tôt vers un amalgame entre intelligence et conscience, avec des arguments intéressants mis en avant par des chercheurs comme John Searle. L’argument de la Chambre Chinoise présenté par Searle [Searle, 1980] (Chinese Room Argument ou CRA) n’est pas une objection contre l’argument principal de Turing, qui possède ses propres vertus malgré de nombreuses limitations, mais c’est un appel du pied pour reconnaître le niveau humain de la conscience déduit du comportement d’intelligence en général. De plus, la philosophie de l’esprit est passée du matérialisme au fonctionnalisme, puis au computationnalisme [Dodig-Crnkovic, 2007], mais peu de choses ont été faites au niveau de la mise en relation – conceptuellement et techniquement – entre la computation comme modèle de conscience et la cognition, jusqu’à très récemment.

On dispose aujourd’hui d’une théorie formelle controversée de la conscience, peut-être vraie ou fausse mais qui a le mérite d’être précise et bien définie, même si elle a évolué sur une courte période de temps. Ce qu’on

appelle la Théorie Intégrée de l'Information [Oizumi et al., 2014] fournit d'intéressants fondements basés sur la computation et la théorie de l'information quant aux conditions nécessaires pour la conscience. Elle propose une mesure d'information intégrée entre des éléments interagissant qui tiennent compte de ce qui est essentiel pour une conscience, c-à-d la sensation d'une expérience interne, et donc la génération d'une information à l'intérieur du système, excédant l'information reçue extérieurement par le système, et indépendante de ce que le système récupère, s'il récupère quelque chose.

De plus, on dispose aujourd'hui d'une théorie formelle capable de prendre en compte des biais de la perception humaine que la théorie classique des probabilités ne peut que quantifier mais pas expliquer pas plus qu'elle ne peut fournir sur eux une hypothèse de travail [Gauvrit et al., 2014c, Gauvrit et al., 2014b]. Ce modèle algorithmique donne une relation forte et formelle entre cognition et computation grâce à une récursion qui semble être innée ou qu'on développe au cours de nos vies [Gauvrit et al., 2014b]. Il faut savoir aussi que cette théorie n'est pas insensible aux arguments de la super-computation de Turing [Zenil and Hernandez-Quiroz, 2007], qui suggèrent que, bien que les mécanismes concrets de cognition ainsi que l'esprit puissent être très différents des opérations d'une machine de Turing, leur puissance computationnelle peut relever de l'Universalité de Turing. Mais on ne sait toujours pas avec certitude quelles seraient les conditions nécessaires ou suffisantes pour tenir compte algorithmiquement des mêmes biais de l'esprit impliquant plus ou moins de puissance computationnelle, et c'est une orientation intéressante pour des investigations futures (plus précisément, on a besoin de certifier si des théorèmes comme le théorème du codage algorithmique restent valides sous des conditions de super ou sous-universalité).

Les connexions suggérées entre computation et vie vont bien au-delà de la cognition. Sidney Brenner, un des pères de la biologie moléculaire, défend l'idée que les machines de Turing et les cellules ont beaucoup en commun [Brenner, 2012]. Plus récemment on a montré lors d'expériences systématiques avec la levure, que les chemins de l'évolution conduisent par des mutations complètement différentes sous des environnements équivalents, vers le même résultat d'évolution [Kryazhimskiy et al., 2014], ce qui pourrait confirmer une autre propriété computationnelle connue sous le nom de confluence dans les systèmes abstraits de réécriture, connue aussi comme la propriété de Church-Rosser ou la propriété du diamant. Ce nom vient d'un résultat connu comme étant le théorème de Church-Rosser, du nom des scientifiques qui l'ont prouvé [Church and Rosser, 1936], établissant que lorsqu'on applique des règles de réduction à des termes en lambda-calcul, un langage équivalent aux machines de Turing pour ce qui est de l'expression et de la puissance de

calcul, l'ordre selon lequel les réductions sont choisies ne fait pas de différence pour le résultat final. Ce type particulier de contingence rend l'évolution plus prévisible qu'on ne l'attendrait, mais cela signifie aussi que trouver les chemins conduisant à une maladie, telle qu'une maladie neurodégénérative, peut s'avérer plus difficile à caractériser à cause de la diversité des causes possibles conduisant au même résultat indésirable. De fortes connexions algorithmiques entre le comportement animal, la biologie moléculaire et la computation de Turing ont été explorées dans [Zenil et al., 2012b, Zenil and Marshall, 2013]. Mais les connexions entre computation et cognition peuvent être retracées en remontant au tout début du domaine, ce qui va nous aider à poser les bases de ce que nous pensons être une importante contribution vers une meilleure compréhension de la cognition, tout particulièrement la cognition humaine, grâce à l'effet lentille grossissante de l'algorithmie.

9.1.1 Le test de Turing est trivial, mais l'esprit est algorithmique

En principe, réussir le test de Turing est trivialement réalisable par force brute. On peut le démontrer en utilisant une simple expérience de pensée du type argument de la Chambre Chinoise. Prenons le nombre de phrases (compréhensibles) dans une conversation. Ce nombre est fini car l'ensemble fini des phrases compréhensibles est limité, et donc fini. Remplissons une table à consulter avec toutes les réponses possibles à toute question possible. Réussir le test de Turing est trivialement réalisable en un temps et un espace fini, par force brute ; c'est juste un problème de combinatoire. Les tables de consultation démarrent au temps computationnel 0 (1), mais si une machine réussit le test de Turing par force brute, leurs tailles devrait croître très exponentiellement pour des phrases croissant, elles, seulement de façon linéaire, étant donné la large étendue des réponses possibles. Réussir le test de Turing, même pour des phrases et des conversations courtes, demanderait plus d'espace et de temps que l'âge et la taille de l'univers observable. Par exemple, la longueur moyenne des phrases du roman *Orgueil et Préjugés* de Jane Austen est de 19 mots. Même pour des questions plus courtes, disons 10 mots, cela implique un nombre de 10^{100000} combinaisons possibles de phrases, où 100000 est un chiffre bas pour le nombre moyen de mots dans un petit dictionnaire ordinaire. Et même en descendant à 10000, le nombre de combinaisons est déjà d'une taille considérable. (1) Coder de telles tables de consultation demanderait un nombre de bits plus grand que le nombre d'atomes dans l'univers (qu'on suppose compris entre 10^{78} et 10^{82} [Lloyd, 2002]), et même si on y arrivait, alors (2) accéder à une telle table géante prendrait aussi plus

de temps que l'âge de l'univers.

En effet, Lloyd estime que l'univers n'a pas pu réaliser plus de 10^{120} opérations sur 10^{90} bits [Lloyd, 2002]. Cela signifierait que tout cerveau humain soit capable d'encoder plus de données que la quantité de données pouvant être stockée en utilisant toutes les particules élémentaires de l'univers ; et que l'esprit soit capable d'accéder à ces données en un temps appelé temps réel, assez vite pour que ce soit en temps réel pour nous. D'après les arguments (1) et (2), l'esprit ne peut pas implémenter et travailler sur une table de consultation qui soit le modèle de référence pour réussir le test de Turing. Cela ne peut pas être la façon dont l'esprit fonctionne, car comme (1) et (2) le montrent clairement, ce serait un défi ou alors il faudrait faire une supposition hors norme au sujet du cerveau (ce qu'on ne peut pas totalement exclure, voir [Zenil and Hernandez-Quiroz, 2007]) et donc au sujet de l'univers, même avec un petit nombre de mots et un "court" temps d'accès à une table de consultation géante. On peut prendre l'exemple de certaines phrases de nature, en quelque sorte, infinie, et que l'esprit humain peut comprendre.

Construisons une phrase nidifiante en utilisant le mot "signification". "La signification de la signification" est encore une phrase relativement facile à saisir, mais le troisième degré peut cependant déjà être un challenge, et si ce n'est pas le cas, on peut nidifier n fois jusqu'à ce que la phrase paraisse s'échapper au-delà de la compréhension humaine. A un certain niveau, on peut concevoir un grand n pour lequel on puisse encore dire qu'une certaine compréhension humaine est possible. On peut prendre l'exemple d'un n sans limites, sinon infini, à partir duquel l'esprit humain puisse encore en tirer un certain sens, rendant cela impossible à traiter par une table de consultation, à cause de l'absence de limites d'un n toujours croissant. Cela s'accorde avec les boucles auto-référentielles de Hofstadter [Hofstadter, 2008] . Voyez la phrase "Quelle est la signification de cette déclaration ?" qu'on peut nidifier plusieurs fois, la transformant en énigme de laquelle l'esprit humain, après un nombre arbitraire de degré de transformation nidifiante, puisse encore être capable d'en tirer quelque chose, même si c'est compris de travers, et peut-être même qu'en s'effondrant sur elle-même, elle crée une espèce de boucle étrange où plus grands sont les n , moins il y a de contenu informatif absolu dans la phrase, sans que ce soit uniquement lié à la taille, d'où un sens additionnel se rapprochant de 0 pour chaque n toujours croissant, d'où une signification finie tirée d'une phrase infinie asymptotiquement nidifiée. Comprendre ce genre de phrase de la façon dont c'est décrit, semble demander une véritable conscience de la nature du contexte dans lequel ces phrases sont construites, et donc nécessite des capacités propres d'une profonde abstraction. C'est le cas célèbre des phrases du type paradoxe-du-menteur, telle que "Cette

déclaration est fausse”, menant finalement à des arguments de type Gödel, pour lesquels on peut atteindre des solutions en se plaçant hors du système en question. Cela a permis à Gödel de finalement déconnecter preuve et vérité grâce à ses théorèmes d’incomplétude, montrant qu’en effet, une certaine auto-conscience du problème est nécessaire pour se rendre compte en tant qu’observateur extérieur – de la vérité d’un fait ne pouvant pas être prouvé à l’intérieur de la théorie, il y a donc un intérieur et un extérieur.

Il est vrai que toutes les combinaisons de mots ne forment pas des phrases grammaticalement valides, donc passer le test de Turing par la force brute nécessite la capacité à reconnaître ces phrases invalides, de façon à ne pas avoir à chercher de réponses pour elles. Ce nombre plus bas suppose aussi que pour la même phrase, la table de consultation donne la même réponse, car la même phrase va se voir affubler le même index produit par une fonction de codage. Il ne s’agit encore que d’implémenter une table de consultation sans autre information complémentaire, comme la probabilité pour un mot d’être suivi par un autre, ce qui entraînerait une autre réduction effective du nombre de combinaisons. Mais en supposant une table de consultation sans fioriture et sans implémenter aucun autre algorithme, pour implémenter une sorte de mémoire, il faudrait travailler au niveau des conversations et pas au niveau des phrases ou même des paragraphes, de cette façon le questionneur peut obtenir une réponse raisonnable à la question “Vous rappelez-vous ma première question?”, et être capable de réussir un test de Turing bien conçu. Même si le nombre final de combinaisons de conversations possibles est supérieur de plusieurs ordres de grandeur aux plus grandes magnitudes astronomiques, même si on se sert de nombres les plus optimistes, c’est encore un nombre fini. Cela signifie que, d’un côté, le test de Turing est computationnellement réalisable trivialement car il suffit de construire une table de consultation assez grande pour obtenir une réponse à chaque phrase possible dans une conversation. D’un autre côté, que le test de Turing soit impossible à réussir en pratique par la force brute au moyen d’une table de consultation, mais que l’esprit humain y arrive signifie que l’esprit ne peut pas implémenter de table de consultation et donc il supporte la computation d’une façon différente.

Et c’est justement ce que Searle aide à entrevoir. La compréhension pour lui ne peut pas se réaliser par la force brute d’une table de consultation, et c’est à quoi il pense quand il parle de programmes informatiques. Mais tous les logiciels ne sont pas des tables de consultation. Les objections au test de Turing peuvent être de type métaphysique, ou de type conforme à l’objection de Searle en introduisant la contrainte des ressources, ou d’autres types d’options non évoqués ici. Cela veut dire que, ou bien l’esprit possède

certaines propriétés métaphysiques non représentables et non reproductibles par la science, ou bien le test de Turing et donc l'opération computationnelle de l'esprit ne peuvent faire sens que si on tient compte des ressources. Ce qui revient à dire que le test de Turing doit se produire en utilisant une certaine quantité limitée d'espace et de temps, sinon le test n'est pas un indice de l'intelligence des machines. Sinon l'intelligence des machines est déconnectée du test de Turing (et de la computation), et doit se comprendre comme une forme à la fois de règles et de données de compression/décompression. Searle a raison en disant que le cerveau ne fonctionne sûrement pas comme un logiciel travaillant sur une colossale table de consultation, et la bonne réponse peut se résumer par une citation de Chaitin : "la compression est la compréhension" [Chaitin, 2006].

L'opposition de Searle aux programmes du genre table de consultation s'incarne dans l'Argument de la Chambre Chinoise (Chinese Room Argument ou CRA), mais le CRA n'est pas représentatif de tous les programmes informatiques, car Searle n'est pas un métaphysicien (il est d'accord). C'est encore une fois profondément lié à de la computation, plus précisément à des questions de complexité computationnelle et algorithmique (taille du programme) ainsi qu'à des questions de complexité purement computationnelles (ressources). A la lumière de la trivialité théorique de la réussite au test de Turing, S. Aaronson souligne le besoin de poser la question des ressources [Aaronson, 2012]. Dans le cadre de l'hypothèse d'un esprit computationnel, et en tirant les conséquences des arguments de Searle, la considération des ressources suggère par exemple que la distinction de classes de complexité temporelle possède une certaine signification métaphysique. Un logiciel de taille exponentielle qui réussit le test de Turing ne serait pas forcément doué de sensibilité (ou doué de conscience, intelligent ou conscient de lui-même), mais par contre un logiciel limité à une taille polynomiale, avec exactement le même comportement entrée/sortie que le logiciel exponentiellement limité, serait doué de sensibilité.

Cela signifie que l'esprit implémente des règles, qui sont d'une certaine façon fortement algorithmiques par nature, et qui compressent à tour de bras de grandes quantités d'informations de manière(s) efficace(s). Nous avons conçu et bâti certains tests basés sur cette supposition forte mais raisonnable, en utilisant des méthodes tirées de la complexité algorithmique et des probabilités algorithmiques, pour étudier l'esprit.

Par l'intermédiaire de la compression, on peut ainsi établir un lien intéressant avec l'Information Intégrée. Imaginez que vous ayez deux fichiers, avec chacun deux versions : compressée et non compressée. L'un est incompressible, et donc tout changement dans sa version non compressée peut sim-

plement être reproduit dans sa version compressée exactement de la même façon, car les deux versions sont effectivement les mêmes. Car tous les bits sont indépendants les uns des autres dans les deux versions, non compressée et compressée. Par contre, pour un fichier compressible, pour reproduire tout changement conduisant à un effet dans sa version non compressée, il va falloir un changement crypté dans la version compressée afin de reproduire le même résultat non compressé, car l'algorithme de compression tire avantage des régularités courtes et longues, c-à-d qu'il utilise la non-indépendance des bits, qui peuvent être encodés dans une seule région du fichier compressé, mais peuvent être très éloignés les uns des autres dans la version non compressée. Cela signifie que les fichiers incompressibles possèdent peu d'information intégrée, alors que les fichiers compressibles possèdent un degré plus grand d'information intégrée. Des idées similaires sont explorées dans [Maguire et al.,], argumentant contre la théorie de l'information intégrée en disant qu'elle n'est pas conforme avec le concept intuitif de conscience. Car ce que les gens entendent par le terme "conscience", c'est qu'un système conscient ne peut pas être analysé et décomposé, parce que si c'était le cas, alors il ne relèverait pas de la "conscience".

Les moteurs de recherche et le web constituent en pratique une sorte de table de consultation exactement du type attaqué par l'argument du CRA (Chambre Chinoise) de Searle. En effet, la probabilité de trouver une page web contenant une permutation de mots représentant une courte phrase est haute, et tend à augmenter avec le temps même si la probabilité croît de façon lente mais exponentielle à cause de l'explosion combinatoire. Le web contient 4,73 milliards de pages environ (<http://www.worldwidewebsite.com/> comme estimé et daté du mercredi 7 janvier 2015), avec des textes surtout écrits par des humains. Mais que le web vu à travers un moteur de recherche soit un genre de table de consultation, est un argument contre l'idée que le web soit un genre d'"esprit global" en accord avec l'argument de Searle. En effet, il y a très peu d'assurance que quelque chose dans le web soit sous une certaine forme compressée (ce n'est pas la même chose avec internet en général, où de nombreuses transactions et protocoles implémentent de la compression d'une façon ou d'une autre, par ex. par encryptage). Nous proposons donc que la compression soit une condition nécessaire pour les esprits et les consciences, mais clairement pas suffisante (voir par ex. les fichiers compressés comme le sont les images au format PNG dans le disque dur d'un ordinateur).

Il s'est avéré dans des expériences de terrain que les animaux disposent de fortes tendances comportementales. Par exemple, dans la communication de Reznikova sur des expériences avec des fourmis, des instructions simples

furent communiquées plus vite et plus efficacement que des informations complexes par des fourmis éclaireurs recherchant des îlots de nourriture localisés aléatoirement dans un labyrinthe. Ces séquences de communication se composent de mouvements droite/gauche codables en binaire (R et L) (voir <http://grahamshawcross.com/2014/07/26/counting-ants/> Consulté le 23 Décembre 2014) [Reznikova and Ryabko, 2011, Reznikova and Ryabko, 2012, Ryabko and Reznikova, 2009] qui sont communiquées par des fourmis éclaireurs retournant de leur équipée de recherche de nourriture dans la colonie pour communiquer des instructions afin d’atteindre cette nourriture. Nous avons quantifié certaines de ces expériences de comportement animal en confirmant par des résultats positifs le soupçon des auteurs quant à la possibilité de rendre compte de ces tendances à l’aide de la théorie de l’information algorithmique [Zenil, 2015].

Les humains aussi disposent de tendances d’un même genre algorithmique, allant de leurs trajectoires de déplacement [Peng et al., 2014] jusqu’à leurs perceptions de la réalité [Chater, 1999]. En effet, nous avons montré que la cognition, incluant la perception visuelle et la création d’aléatoire subjectif, possédait une tendance dont on peut rendre compte avec le concept fertile de probabilité algorithmique. En se servant d’un ordinateur pour observer le comportement humain d’une nouvelle façon, spécialement en utilisant un test de Turing inversé où c’est l’esprit humain qui est analysé, et avec l’aide d’une machine de Turing “moyenne” ou bien un logiciel implémentant tous les algorithmes de compression possibles, nous allons montrer que l’esprit humain se comporte plutôt comme une machine. Nous allons effectivement inverser la question originale de Turing, qui était posée via son jeu d’imitation destiné à savoir si les machines se comportent comme nous.

9.2 La complexité algorithmique comme modèle de l’esprit

Depuis l’émergence du paradigme bayésien en sciences cognitives, les chercheurs ont exprimé le besoin d’une prise en compte formelle de la complexité basée sur une mesure valable de la complexité. Ils se sont efforcés aussi de trouver un moyen de donner un compte-rendu formel et normatif de la probabilité qu’un algorithme déterministe produise une séquence donnée, comme la chaîne pile-ou-face “HTHTHTHT”, qui ressemble intuitivement à un résultat de processus déterministe même si elle a la même probabilité d’apparition que n’importe quelle autre séquence de même longueur selon la théorie classique

des probabilités, qui assigne une probabilité de $1/2^8$ à toutes les séquences de taille 8.

9.2.1 Des tendances à Bayes

Parmi les divers domaines des sciences cognitives ayant exprimé le besoin d’une nouvelle mesure de complexité, le plus motivé est le domaine du raisonnement probabiliste. Le fameux travail de Kahneman, Slovic et Tversky (1982) [Tversky and Kahneman, 1974] visait à comprendre comment les gens raisonnent et prennent des décisions face à des sources d’incertitude et de bruit informatif. Ils ont montré que les humains avaient tendance à faire de nombreuses erreurs sur l’aléatoire et les probabilités. Par exemple, on a tendance à prétendre que la séquence tirée à pile-ou-face “HT-THTHHHTT” a plus de probabilité d’apparaître au lancer de pièces, que la séquence “HHHHHTTTTT”.

Dans l’approche “heuristiques et tendances” défendue par Kahneman et Tversky, ces erreurs “systématiques” ont été interprétées comme inhérentes à la psychologie humaine, ou alors comme le résultat de l’utilisation d’une heuristique défectueuse. Par exemple, on croyait que les gens avaient tendance à dire que “HHHHHTTTTT” était moins aléatoire que “HTTHTHHHTT” parce qu’ils étaient influencés par une heuristique dite de représentativité, selon laquelle une séquence est plus aléatoire quand elle est en meilleure conformité avec des exemples prototypiques de séquences aléatoires. On a défendu l’idée que le raisonnement humain fonctionnait comme un ordinateur défectueux. Bien que de nombreux articles aient été publiés sur ces biais, on ne sait pas grand-chose sur leurs causes. Un autre exemple sur une erreur largement répandue est ce qu’on appelle la “tendance à l’équiprobabilité” [Lecoutre, 1992], une tendance à croire que toute variable aléatoire devrait être uniforme (avec une probabilité égale pour toutes les formes possibles que cette variable va prendre). Dans le même veine que les investigations de Kahneman et al. [Tversky and Kahneman, 1974], ce biais aussi, c-à-d la supposition erronée que l’aléatoire implique l’uniformité, a depuis longtemps été interprétée comme un faille fondamentale de l’esprit humain.

Durant les quelques dernières dizaines d’années, un changement de paradigme s’est produit en sciences cognitives. Ce “nouveau paradigme” – ou approche bayésienne – suggère que l’esprit humain (ou animal) n’est pas une machine défectueuse. mais une machine probabiliste d’un certain type. Selon cette façon d’appréhender la cognition humaine, nous tous estimons et révisons constamment les probabilités des événements du monde qui nous en-

ture, en prenant en compte toute nouvelle bribe d'information, et en suivant plus ou moins des règles probabilistes (dont les bayésiennes).

Les études dans cette direction essayent souvent d'expliquer nos erreurs probabilistes en termes d'intuition correcte sur l'aléatoire ou sur une probabilité mais appliquée dans un contexte inapproprié. Par exemple, une ré-analyse mathématique et psychologique du biais d'équiprobabilité fut récemment publiée [Gauvrit and Morsanyi, 2014]. La théorie mathématique de l'aléatoire, basée sur la complexité algorithmique (ou sur l'entropie, selon le cas), implique en fait de l'uniformité. Donc prétendre que l'intuition associant aléatoire et uniformité est un biais, ne colle pas à la théorie mathématique. D'un autre côté, si on suit la théorie mathématique de l'aléatoire, on doit admettre qu'une combinaison d'éléments aléatoires n'est généralement plus du tout aléatoire. Ainsi le biais d'équiprobabilité (qui effectivement biaise, car il fournit fréquemment des réponses erronées en termes de classe de probabilités) n'est pas, pensons-nous, le résultat d'une erreur de conception par rapport à l'aléatoire, mais plutôt une conséquence de l'intuition incorrecte que des événements aléatoires peuvent se combiner sans affecter leurs propriétés d'être aléatoires.

Les heuristiques de représentativité ont elles aussi été ré-analysées à la lumière du nouveau paradigme bayésien. Les chercheurs pensent maintenant que lorsqu'on doit comparer la probabilité qu'une pièce non-truquée produise "HHHHHTTTTT" avec la probabilité de n'importe quelle autre séquence de 10 items, cela ne se passe pas vraiment ainsi. Une des raisons est que cette question n'est pas naturelle : notre cerveau est construit pour estimer la probabilité des causes des événements observés, et pas les probabilités a priori de ces événements. Donc, disent les chercheurs, quand on a des participants qui doivent doser la probabilité de la chaîne s "HHHHHTTTTT" par exemple (ou toute autre chaîne), ils n'estiment pas la probabilité qu'une telle chaîne apparaisse lors d'un enchaînement de pile-ou-face, ce qui pourrait s'écrire comme $P(s|R)$ où R vaut pour "processus aléatoire", mais plutôt la probabilité inverse $P(R|s)$, qui est la probabilité que la pièce de monnaie soit non-truquée (ou que la chaîne soit véritablement aléatoire), par rapport au s produit par cette pièce. Une telle probabilité peut s'estimer en se servant du théorème de Bayes :

$$P(R|s) = \frac{P(s|R)P(R)}{P(s|R)P(R) + P(s|D)P(D)}$$

où D vaut pour "non-aléatoire" (ou déterministe). Dans cette équation, le seul élément problématique est $P(s|D)$, la probabilité qu'un algorithme

indéterminé mais déterministe, va produire s . On a longtemps cru qu’aucune mesure normative de cette valeur ne pourrait se supposer, bien que certains auteurs aient eu l’intuition que c’était lié à la complexité de s : de simples chaînes vont avoir plus de chances d’apparaître comme résultat d’un algorithme, plutôt que des chaînes complexes.

La théorie algorithmique de l’information fournit concrètement un cadre formel pour cette intuition. La probabilité algorithmique d’une chaîne s est la probabilité qu’un programme choisi au hasard, et lancé sur une Machine de Turing Universelle (à préfixe libre), va produire s pour ensuite s’arrêter. On va donc s’en servir comme définition formelle naturelle de $P(s|D)$. Comme on va le voir plus loin, la probabilité algorithmique d’une chaîne s est inversement liée à sa complexité algorithmique (de Kolmogorov-Chaitin), définie comme la longueur du plus court programme produisant s et s’arrêtant : les chaînes simples possèdent une plus haute probabilité algorithmique.

Un inconvénient important de la complexité algorithmique est qu’elle n’est pas calculable. Cependant, il existe aujourd’hui des méthodes pour approximer cette probabilité, et ainsi la complexité, de toute chaîne, même les courtes (voir ci-dessous), ceci suscitant un regain d’intérêt pour la complexité dans les sciences cognitives.

Par ces méthodes, on peut calculer que, avec une probabilité à priori de 0.5, la probabilité que la chaîne HHHHHTTTTT soit aléatoire vaut 0.58, alors que la probabilité que HTTHTHHHTT soit aléatoire vaut 0.83, confirmant ainsi l’intuition commune que cette dernière est “plus aléatoire” que la première.

9.2.2 La méthode du théorème du Codage

Une méthode pour évaluer la complexité de Kolmogorov-Chaitin, appelée méthode par compression sans perte, incarnée par l’algorithme de Lempel-Ziv, est utilisée largement et depuis longtemps. Ce genre d’outil, associé à l’entropie classique de Shannon [Wang et al., 2014], a servi récemment en neuropsychologie pour explorer la complexité des données EEG ou fMRI (imagerie fonctionnelle par résonance magnétique) [Maguire et al., , Casali et al., 2013]. En effet, la taille d’un fichier compressé donne une indication sur sa complexité algorithmique. D’une part, les méthodes de compression possèdent un inconvénient de base : elles ne peuvent reconnaître que les régularités statistiques et sont donc des implémentations des variations de mesures entropiques, évaluant seulement le taux de convergence entropique en se basant sur des répétitions de chaînes dans des fenêtres coulissantes de

taille fixe. Si les algorithmes de compression sans perte fonctionnent pour estimer la complexité de Kolmogorov, c’est uniquement parce que la compression est un test suffisant de non-aléatoire, mais il y a une autre direction où ils échouent clairement, c’est de savoir avec certitude si quelque chose est le résultat d’un processus algorithmique (par ex. les décimales de la constante mathématique π). Ce qui fait qu’ils ne peuvent pas détecter les structures prenant d’autres formes que la répétition simple. D’autre part, les méthodes par compression ne sont pas appropriées pour les courtes chaînes (disons moins de quelques centaines de symboles). Pour les courtes chaînes, les algorithmes de compression donnent souvent des fichiers résultats plus longs que les chaînes elles-mêmes, d’où la production de résultats très instables et difficiles, sinon impossibles à interpréter.

En sciences cognitives et expérimentales, les chercheurs ont l’habitude de traiter de courtes chaînes avec tout au plus quelques dizaines de symboles, pour lesquels les algorithmes de compression sont donc mis en défaut. C’est la raison pour laquelle ils se sont tournés plutôt vers des méthodes ‘sur mesure’.

La méthode par le théorème du Codage [Delahaye and Zenil, 2012, Soler-Toscano et al., 2014] a été spécifiquement conçue pour résoudre ce problème. Grâce au théorème du Codage, les chercheurs ont produit des valeurs pour la “complexité algorithmique dans les chaînes courtes” (qu’on va noter en abrégé ACSS ou Algorithmic Complexity for Short Strings). ACSS est disponible gratuitement en tant que R-package (nommé ACSS), ou par un calculateur de complexité en ligne (www.complexitycalculator.com Consulté le 26 Décembre 2014) [Gauvrit et al., 2014c]. A la base de l’ACSS, il y a l’idée que la probabilité algorithmique peut servir à saisir la complexité algorithmique. La probabilité algorithmique d’une chaîne s est définie comme la probabilité qu’une machine de Turing U universelle à préfixe libre, va produire s puis s’arrêter. Formellement

$$m(s) = \sum_{U(p)=s} \frac{1}{2^{-|p|}}$$

La complexité algorithmique d’une chaîne s est définie comme la longueur du plus court programme qui, lancé sur une machine de Turing U universelle et à préfixe libre, va produire s puis s’arrêter

$$K(s) = \min(|p| : U(p) = s)$$

$K(s)$ et $m(s)$ dépendent tous deux du choix de la machine U . Donc l’expression “la complexité algorithmique de s ” est en elle-même un raccourci.

Pour les chaînes longues, cette dépendance est relativement faible. En effet, le théorème d’invariance établit que pour n’importe quelles U et U' , deux machines de Turing universelles et à préfixe libre, il existe une constante c indépendante de s , tel que

$$|K_U(s) - K_{U'}(s)| \leq c$$

La constante c peut être arbitrairement grande. Si on veut avoir une approximation de la complexité algorithmique de chaînes courtes, le choix de U est donc déterminant. Pour dépasser cet inconvénient, on peut tirer avantage d’un lien formel établi entre complexité algorithmique et probabilité algorithmique. Le théorème du Codage algorithmique déclare que

$$K_U(s) = -\log_2(m_U(s)) + O(1)$$

Ce théorème peut servir à donner une approximation de $K(s)$ en passant par une estimation de $m(s)$.

Au lieu de choisir une machine de Turing universelle et arbitraire, particulière, et de l’alimenter avec des programmes aléatoires, Delahaye et Zenil (2012) [Delahaye and Zenil, 2012] ont eu l’idée (de façon formellement équivalente) d’utiliser un très grand échantillonnage de machines de Turing et de les lancer sur des rubans vides. En agissant ainsi, ils ont construit une distribution expérimentale “naturelle” (une énumération de type dictionnaire) approchant $m(s)$, conçue comme un $m_U(s)$ moyen. Ils ont alors défini $ACSS(s)$ comme $-\log_2(m(s))$. $ACSS(s)$ est une approximation de $K_U(s)$. Pour valider la méthode, les chercheurs ont étudié comment $ACSS$ varie sous différentes conditions. On a trouvé que $ACSS$ calculé avec différents très grands échantillonnages de petites machines de Turing restait stable [Gauvrit et al., 2014c]. De même, plusieurs définitions de machines de Turing n’altèrent pas $ACSS$ [Delahaye and Zenil, 2011]. En utilisant des automates cellulaires au lieu des machines de Turing, ils ont montré que $ACSS$ restait relativement stable. A un niveau plus pratique, $ACSS$ est aussi validé par des résultats expérimentaux. Par exemple, comme on va le voir ci-dessous, $ACSS$ est relié à la perception humaine de la complexité. Et on a trouvé des applications en théorie des graphes et en biologie des réseaux [Zenil et al., 2014].

9.2.3 La Méthode de Décomposition par Blocs

ACSS fournit aux chercheurs un outil convivial pour estimer la complexité algorithmique des très courtes chaînes. Malgré l'énorme échantillon (des milliards) de machines de Turing utilisé pour construire $m(s)$ et $K(s)$ en se servant de la méthode du théorème du Codage, deux chaînes de longueur 12 ne furent jamais produites, mais on peut utiliser ACSS avec confiance pour des chaînes allant jusqu'à 12 en assignant aux deux chaînes manquantes la plus grande complexité de l'ensemble plus 1.

La même méthode utilisée avec des machines de Turing bidimensionnelles produit un échantillonnage de structures binaires sur des grilles, mais d'une portée limitée ici encore. Toutes les grilles 5 x 5 ne sont pas présentes dans les résultats, mais on peut faire une approximation de n'importe quelle grille $n \times n$ en la décomposant en carrés 4 x 4 dont on a une estimation connue grâce à la méthode bi-dimensionnelle du théorème du Codage. Cette version de la méthode implique simplement de lancer les machines de Turing sur des trames plutôt que des rubans, ensuite toute la théorie autour du théorème du Codage peut s'appliquer et on peut concevoir une méthode bi-dimensionnelle du théorème du Codage.

Cette idée de décomposer va combler le trou entre la zone d'application des méthodes classiques de compression sans perte (longues chaînes) et la zone d'application de l'ACSS (courtes chaînes). Dans ce but, une nouvelle méthode basée sur l'ACSS a été développée : la méthode de décomposition par Blocs [Zenil et al., 2014]. Au départ, l'idée de base de la décomposition par Blocs est de décomposer des chaînes dont on possède les approximations exactes de leur complexité de Kolmogorov, en sous-chaînes plus courtes (avec des chevauchements possibles). Par exemple la chaîne "123123456" peut se décomposer en sous-séquences de 6 symboles avec un chevauchement de 3 symboles "123123" et "123456". Ensuite, la méthode de décomposition par blocs tire avantage des propriétés connues issues de la théorie de l'information, en pénalisant n répétitions, ce qui peut se traduire en utilisant seulement $\log_2(n)$ bits par la formule :

$$C(s) = \sum_p \log_2(n_p) + K(p)$$

où p représente les différents types de sous-chaînes, n_p le nombre d'occurrences de chaque p , et $K(p)$ la complexité de p approchée par ACSS. Comme le montre la formule, la méthode de décomposition par Blocs tient compte à la fois de la complexité locale de la chaîne, et à la fois des répétitions (même sur

de longues distances). Avec la méthode du théorème du Codage, la méthode de décomposition par blocs et les algorithmes de compression, la palette totale des longueurs de chaînes peut maintenant être couverte avec une famille de méthodes formelles. A l'intérieur du champ des sciences cognitives et comportementales, ces nouveaux outils sont d'une pertinence très concrète. Dans la section suivante, nous allons décrire trois domaines récents des sciences cognitives dans lesquels la complexité algorithmique joue un rôle important. Dans ces domaines, ACSS et les méthodes décomposition par Blocs ont joué un rôle-clé dans l'approximation de la complexité algorithmique de chaînes ou de formations de courtes longueurs (2-12) et de moyennes longueurs (10-100).

9.3 Cognition et Complexité

Un des premiers exemples d'une application vers la cognition, du théorème du Codage et des méthodes de décomposition par Blocs, a permis de quantifier la complexité de séquences courtes dans les expériences de Reznikova sur la communication des fourmis, et de valider leurs résultats [Zenil, 2013a]. En effet, on a trouvé que les fourmis communiquaient des chaînes plus courtes (par rapport aux instructions permettant d'aller chercher de la nourriture dans un labyrinthe) dans un temps plus court aussi, établissant ainsi un lien expérimental entre le comportement animal, la complexité algorithmique et la complexité temporelle.

L'idée fut développée pour établir une relation entre contenu informatif et énergie (dépensée à la fois pour fourrager et pour communiquer), pour finalement établir un paysage d'adaptation basé sur les seuils entre ces valeurs, comme reporté dans [Zenil et al., 2012b]. Comme cela est montré dans [Zenil et al., 2012b], si l'environnement est trop prévisible, le coût du traitement de l'information est très bas. A l'opposé, si l'environnement est aléatoire, le coût est maximum. Les résultats obtenus par ces méthodes de traitement de l'information suggèrent que les organismes dotés de meilleures capacités d'apprentissage économisent plus d'énergie et ont donc un avantage évolutif.

De bien des façons, le comportement animal (incluant notamment le comportement humain) laisse supposer que le cerveau agit comme un système de compression de données. Par exemple, malgré notre mémoire très limitée, il est clair que nous pouvons retenir de longues chaînes si elles ont une complexité algorithmique faible (par ex. la série 123456... contrairement aux décimales de π , voir ci-dessous). Les sciences cognitives et comportementales ont affaire la plupart du temps à de petites séquences, dépassant rarement

quelques dizaines de valeurs. Pour de telles séquences courtes, estimer la complexité algorithmique est un défi. En effet, jusqu’à maintenant, les sciences comportementales se basaient largement sur une prise en compte intuitive et subjective de la complexité (par ex. les expériences de Reznikova sur la communication des fourmis). Grâce à la “méthode du théorème du Codage”, il est maintenant possible d’obtenir une approximation fiable de la complexité algorithmique de chaînes de n’importe quelle longueur [Gauvrit et al., 2014c] et ces méthodes ont été mises en test et appliquées pour valider plusieurs intuitions soulignées par d’éminents chercheurs en sciences cognitives au cours des quelques dizaines d’années passées. On va décrire certaines de ces intuitions et expliquer comment elles ont été confirmées par des études expérimentales.

9.3.1 Mémoire de travail

Les chercheurs en psychologie ont identifié différents types de mémoires chez les hommes et autres animaux. Deux genres associés de mémoires ont été particulièrement bien étudiés : la mémoire à court-terme et la mémoire de travail. La mémoire à court-terme fait référence à une sorte de ressource cognitive qui permet le stockage d’information pendant quelques secondes ou minutes. Selon une récente contribution au sujet de la mémoire à court-terme [Barrouillet et al., 2009], les items stockés dans la mémoire à court-terme se désintègrent rapidement et automatiquement quand le temps passe, comme simple résultat de l’écoulement du temps, à moins que l’attention soit consacrée à la réactivation de l’item à-mémoriser. Ainsi une constante réactivation est nécessaire pour garder l’information stockée quelques minutes. Ceci est réalisé au moyen d’une boucle interne de langage (on peut continuellement répéter les items à-mémoriser) ou en utilisant une imagerie mentale (on peut produire des images mentales des items). D’un autre côté, certains chercheurs croient que la désintégration observée dans la mémoire est entièrement due à l’interférence avec une nouvelle information [Oberauer et al., 2004]. Les chercheurs sont donc en désaccord sur les causes tandis qu’ils reconnaissent qu’il y a effectivement une désintégration rapide des traces mémorielles. L’empan de la mémoire à court-terme est estimé à environ 7 items [Miller, 1956, Edin et al., 2009]. Pour arriver à cette estimation, les psychologues utilisent l’empan de tâches de mémorisation simple où on demande aux participants de répéter des séries d’items de longueur croissante. Par exemple, l’expérimentateur dit “3, 5” et les participants doivent répéter “3, 5”. Puis l’expérimentateur dit “6, 2, 9” à haute voix, ce qui est répété par les participants. L’empan de la mémoire à court-terme est défini comme la longueur de la plus longue série dont on peut se rappeler. Étonnamment, cet

empan est à peine dépendant du type d'item à mémoriser (lettres, chiffres, mots, etc.).

Cependant, deux observations jettent un éclairage nouveau sur le concept de mémoire à court-terme. La première est l'effet préjudiciable d'une charge cognitive sur l'empan de la mémoire à court-terme. Par exemple, si les humains se rappellent habituellement jusqu'à 7 items dans une situation basique comme décrit ci-dessus, le nombre d'items correctement rappelés chute si un individu doit réaliser une autre tâche en même temps. Par exemple, si un sujet doit répéter "baba" tout en mémorisant des chiffres, il ne va probablement pas arriver à stocker les 7 items [Barrouillet et al., 2004]. Plus les tâches d'interférence sont contraignantes (comme vérifier si des équations sont justes ou générer des lettres aléatoires), plus l'empan va se raccourcir. La seconde observation est que lorsqu'il y a une structure dans la liste à-se-rappeler, on peut retenir plus d'items. Par exemple, il est improbable que quelqu'un puisse mémoriser d'un coup une série comme "3, 5, 4, 8, 2, 1, 1, 9, 4, 5". Cependant, des séries plus longues comme "1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1" vont facilement se mémoriser.

La première observation conduit à la notion de mémoire de travail. la mémoire de travail est une ressource cognitive hypothétique utilisée à la fois pour stocker et traiter l'information. Quand les participants doivent se rappeler des séries tout en réalisant une deuxième tâche, ils attribuent une partie de leur mémoire de travail à cette deuxième tâche, ce qui réduit la part de mémoire de travail allouée au stockage, c-à-d à la mémoire à court-terme [Baddeley, 1992].

La seconde observation conduit à la notion de découpage en morceaux. On pense aujourd'hui que les humains peuvent diviser des listes à-se-rappeler en sous-listes simples. Par exemple, quand ils ont à stocker "1, 2, 3, 4, 5, 3, 3, 1, 1, 1", ils peuvent construire trois "morceaux" : "1, 2, 3, 4, 5", "3, 3, 3" et "1, 1, 1". Comme cet exemple l'illustre, les regroupements ne sont pas arbitraires : ils sont conçus pour minimiser la complexité, et agissent ainsi comme des algorithmes de compression [Mathy and Feldman, 2012]. Un facteur objectif montrant que le découpage en morceaux se produit est qu'en se remémorant les séries ci-dessus, les gens marquent des pauses plus longues entre les regroupements supposés qu'à l'intérieur de ces regroupements. En prenant en compte ces nouveaux concepts, on pense aujourd'hui que l'empan de la mémoire à court-terme n'est pas d'environ 7 items, comme cela fut établi en se servant de listes non structurées, mais plus précisément d'environ 4 morceaux [Cowan, 2001].

L'empan de la mémoire à court-terme est donc dépendant du type de liste

à mémoriser. Avec des séries structurées, on peut se rappeler de plus d'items, mais de moins de morceaux. Cette apparente contradiction peut se dépasser en dissociant clairement l'empan de la mémoire à court-terme de l'empan de la mémoire de travail. Pour dépasser les limitations de la mémoire à court-terme, les humains peuvent tirer profit de la structure présente dans certaines séries. Agir ainsi nécessite l'utilisation de la composante "traitement" de la mémoire de travail afin d'analyser les données [Mathy and Feldman, 2012]. Ainsi, même des tâches simples où les sujets n'ont qu'à stocker et se rappeler des séries d'items, peuvent éprouver la partie traitement de la mémoire de travail.

Selon cette hypothèse, la mémoire de travail fonctionne comme un algorithme de compression, où une partie de la ressource est allouée au stockage des données, tandis qu'une autre partie est dédiée au programme de "compression/décompression". Des études récentes s'alignent parfaitement avec cette idée. [Mathy et al., 2014] ont utilisé un Simon®, un jeu populaire des années 1980 dans lequel on doit reproduire des séries de couleurs (choisies parmi un ensemble de 4 couleurs) de plus en plus longues, reproduisant les tâches classiques d'une mémoire à court-terme. Ils ont montré que la complexité algorithmique de la série est un meilleur outil de prédiction d'une remémoration correcte, que la longueur de la chaîne. De plus, quand les participants font des erreurs, ils génèrent en moyenne une chaîne (incorrecte) d'une complexité algorithmique plus basse que la chaîne à se rappeler.

Tous ces résultats expérimentaux suggèrent que la mémoire de travail agit comme un algorithme de compression. Dans des conditions normales, la compression est sans perte, mais quand les items à se rappeler excèdent la capacité de la mémoire de travail, il se peut qu'à la place, on utilise de la compression avec perte.

9.3.2 Perception de l'aléatoire

Les humains partagent certaines notions intuitives et apparemment innées (ou au moins précoces) au sujet de faits simples en mathématiques, biologie, physique et psychologie. On trouve cette connaissance de base [Spelke and Kinzler, 2007] chez les bébés âgés de quelques mois. Pour ce qui est du champ des mathématiques simples, par exemple, on a montré que les gens partageaient un certain savoir collectif ou des intuitions sur les nombres et la géométrie. Les humains et autres animaux manifestent en commun l'aptitude à discriminer des quantités en se basant sur un "Sens de l'Estimation des Nombres", grâce auquel on peut immédiatement sen-

tir que 3 objets sont plus que 2, ou 12 plus que 6 [Dehaene et al., 2006]. Parce que ce sens des nombres n'est qu'approximatif, il ne nous permet pas de faire la différence entre 12 et 15, par exemple. Il existe de nombreuses indications que nous possédons un sens inné de la quantité. Par exemple, adultes et enfants sont tous deux capables de "compter" des quantités allant jusqu'à 4 de façon presque instantanée, alors que des quantités plus importantes demandent une stratégie de comptage. Face à 1, 2 ou 3 objets, nous percevons immédiatement combien il y en a sans avoir à compter. Ce phénomène est connu sous le nom de "subitizing" (dénombrement sans comptage) [Mandler and Shebo, 1982]. Les nourrissons âgés de six mois [Xu et al., 2005], les singes [Boysen and Hallberg, 2000] et mêmes les oiseaux [Pepperberg, 2006] sont capables de réaliser des comparaisons basiques entre petits nombres. Cependant, pour des quantités supérieures à 4, nous sommes obligés d'avoir recours à des stratégies apprises comme le comptage.

Les mêmes résultats ont été obtenus dans le champ de la géométrie simple. Dans une étude sur un groupe d'indigènes amazoniens, Dehaene et al. (2006) [Dehaene et al., 2006] a trouvé que les gens sans aucune culture mathématique formelle raisonnaient naturellement en termes de points et de lignes. Ils perçoivent aussi la symétrie et partagent des notions intuitives de topologie, comme la connexité (être d'un seul tenant) ou la connexité simple (d'un seul tenant et sans trous).

Des découvertes récentes en psychologie de l'aléatoire subjectif suggèrent que la connaissance probabiliste pourrait bien figurer sur la liste des savoirs humains de base. Téglás et al. (2011) [Téglás et al., 2011] ont exposé des enfants de 1 an à des images de bocaux contenant différentes proportions de boules de couleur. Quand une boule est sortie au hasard du bocal, on enregistre la durée du regard de l'enfant. On peut facilement aussi calculer la probabilité de l'évènement observé. Par exemple, si une boule blanche est tirée, la probabilité correspondante est la proportion de boules blanches dans le bocal. Il se trouve que les durées des regards des enfants sont corrélés aux probabilités des évènements observés : plus l'évènement est probable, plus le regard sera bref. Ceci est interprété comme une preuve que des enfants âgés de 1 an possèdent déjà un sens intuitif approximatif de la probabilité, de même qu'ils possèdent un sens approximatif des nombres.

Ce sens de la probabilité est même plus riche et complexe que ne le suggère l'exemple précédent. Xu et Garcia (2008) [Xu and Garcia, 2008] ont démontré que des enfants encore plus jeunes (8 mois) pouvaient avoir des comportements montrant une intuition basique de la probabilité bayésienne. Dans leurs expériences, les enfants voyaient un expérimentateur tirer des boules blanches ou rouges d'un bocal opaque. L'échantillonnage résultant pouvait

comporter plus de rouges ou plus de blanches, en fonction du tirage en cours. Une fois l'échantillon prélevé, l'intérieur du bocal était rendu visible, révélant soit une grande proportion de boules rouges, ou une grande proportion de boules blanches. On a observé que les enfants fixaient plus longuement la scène si ce qui s'était produit était moins probable par rapport à une déduction bayésienne inspirée par l'échantillon. Par exemple, si l'échantillon était constitué d'une large majorité de boules rouges, alors que le bocal, une fois visible, montrait un excès de boules blanches, le regard de l'enfant s'attardait plus longtemps sur la scène.

Dans la même veine, Ma et Xu (2009) [Ma and Xu, 2013] ont présenté à des enfants de 9 et 10 mois un bocal contenant autant de boules rouges que de jaunes. Cette fois, l'échantillonnage pouvait être fait par un humain ou par un robot. Ma et Xu montrent que les enfants s'attendent à l'apparition d'une régularité quelconque lors du tirage humain. Cette expérimentation fut conçue pour détecter notre intuition d'intentionnalité qu'on interprète derrière toute régularité. Elle montre aussi que les humains possèdent une intuition précoce de la régularité et de l'aléatoire.

Les adultes sont capables de faire des discriminations plus fines entre des chaînes selon leur complexité. Par exemple, en se servant de tâches classiques de perception d'aléatoire, Matthews (2013) [Matthews, 2013] demandait aux gens de choisir si des chaînes binaires de longueur 21 étaient plus probablement le fruit d'un processus aléatoire ou d'un processus déterministe. Matthews était d'abord intéressé par les effets de contexte associés à de telles tâches. Cependant, il montra aussi que les adultes s'accordent habituellement pour dire quelles chaînes sont plus aléatoires. Une ré-analyse des données de Matthews [Gauvrit et al., 2014a] a montré que la complexité algorithmique est un bon outil de prévision de l'aléatoire subjectif – en l'occurrence, la probabilité qu'un humain considérerait que la chaîne paraît aléatoire.

En essayant de découvrir les sources de notre perception de l'aléatoire, Hsu et al (2010) [Hsu et al., 2010] ont suggéré qu'elles pourraient venir d'un apprentissage du monde. Plus précisément, selon les auteurs, notre idée d'aléatoire bi-dimensionnel pourrait se baser sur la probabilité qu'une structure donnée apparaisse dans une scène visuelle naturelle. Ils ont présenté une série de motifs, composés de 8 cellules noires et 8 cellules blanches sur des grilles 4×4 . Les sujets devaient décider si ces grilles étaient "aléatoires" ou non. La proportion de sujets répondant "aléatoire" a servi de mesure pour l'aléatoire subjectif. Les auteurs ont aussi scanné une série de photos tirées de la nature. Ces images furent binarisées en noir et blanc en utilisant la médiane comme seuil, et toutes leurs grilles 4×4 extraites. A partir de cet ensemble de données, on calcula une distribution. Hsu et al (2010) [Hsu et al., 2010] ont

trouvé que la probabilité qu’une grille apparaisse dans les photos au hasard de scènes naturelles était corrélée à l’estimation humaine : plus fréquente était la grille, moins elle était qualifiée d’aléatoire. Les auteurs interprètent ces résultats comme une démonstration que nous apprenons à percevoir l’aléatoire par nos yeux. Un prolongement de l’étude de Hsu et al (2010) [Hsu et al., 2010] a confirmé la corrélation, et on a trouvé que l’aléatoire subjectif et la fréquence dans les scènes naturelles étaient tous deux corrélés à la complexité algorithmique des motifs [Gauvrit et al., 2014b]. On a découvert que la statistique des scènes naturelles expliquait en partie comment nous percevons l’aléatoire/complexité. On voit ici une distinction nette avec les expérimentations mentionnées précédemment selon lesquelles même des enfants de moins d’un an disposent de la capacité à détecter des régularités, car ces résultats suggèrent que la perception de l’aléatoire n’est pas innée. Notre sens de l’aléatoire peut se développer très tôt dans notre existence, basé sur des scènes visuelles observées et sur un sens inné des statistiques, mais il évolue avec le temps.

Nos résultats dans [Gauvrit et al., 2014b] suggèrent que l’esprit est intrinsèquement câblé pour croire que le monde est algorithmique par nature, pour croire que ce qui se passe en ce monde est probablement le résultat d’un programme de calcul aléatoire et non le résultat d’un processus produisant un aléatoire uniforme classique. Pour savoir si c’est le fait d’un design biologique ou si c’est une “aptitude” développée, on peut regarder si et comment les gens développent cette vision durant leur temps de vie. Des résultats préliminaires suggèrent que cette vision ou ce filtre algorithmique au sujet du monde dont nous sommes équipés, se construit au fil du temps en atteignant un pic de production et de perception algorithmique d’aléatoire vers environ l’âge de 25 ans (voir <http://complexitycalculator.com/hrng/>). Cela signifie que l’esprit s’adapte et apprend à partir d’expériences, et pour une certaine raison il développe une perception algorithmique avant d’en changer encore. Nous n’aurions pas développé un tel pic de notre vision du monde à l’âge de la reproduction, s’il n’y avait pas un avantage évolutif, tout comme il semble improbable que ce soit une caractéristique neutre de l’esprit par rapport au monde, car elle affecte la façon dont nous percevons les événements et donc comment nous en tirons un apprentissage et un comportement. Et cela n’est pas réservé à l’esprit humain, mais touche aussi le comportement animal comme montré dans [Zenil, 2015]. Tout ceci témoigne d’une même direction, que le monde est ou nous apparaît comme hautement algorithmique par nature, au moins provisoirement.

9.3.3 Culture et structure

Dans différents domaines des sciences sociales et des sciences cognitives, on a trouvé que des régularités pouvaient apparaître à partir de la transmission d'items culturels [Smith and Wonnacott, 2010]. Par exemple, dans l'étude des rumeurs, on a découvert que certaines catégories de rumeurs se disséminaient facilement, tandis que d'autres disparaissaient purement et simplement. Les “mèmes” dont on se rappelle facilement, auxquels on croit et qui sont transmis, partagent certaines propriétés de base, comme un équilibre entre les éléments attendus et inattendus, ainsi que la simplicité [Bronner, 2010]. Les histoires trop compliquées, pense-t-on, sont trop difficiles à mémoriser et sont vouées à disparaître. De plus, dans la propagation des rumeurs les transmetteurs font des erreurs, d'une façon partiellement prévisible. Comme une règle, les erreurs rendent les messages plus simples et donc plus faciles à retenir.

Par exemple, Barrett et Nyhof (2001) [Barrett and Nyhof, 2001] avaient des participants qui apprenaient à fabriquer des histoires comportant certains éléments intuitifs et contre-intuitifs. Ils ont trouvé que les éléments légèrement contre-intuitifs étaient mieux retenus que les éléments intuitifs. Dans une recherche selon les mêmes orientations, Atran et Norenzayan (2004) [Atran and Norenzayan, 2004] ont découvert une courbe en U inversé entre la composition d'éléments contre-intuitifs et la mémorisation correcte des histoires.

Deux concepts complémentaires sont importants en psychologie expérimentale de la transmission culturelle. La facilité à être appris mesure la facilité avec laquelle un item (histoire, mot, structure) est mémorisé par quelqu'un. Plus les éléments sont faciles à apprendre, plus ils se disséminent effectivement. Ils sont sélectionnés de façon “darwinienne”, car la facilité à être appris pour un item est l'adaptation de cet item à la communauté humaine dans laquelle il va vivre ou mourir.

La complexité est une autre propriété importante des items culturels. En général, les items complexes sont moins faciles à apprendre, mais il y a des exceptions. Par exemple, une histoire sans intérêt ne va pas être apprise, même si elle est extrêmement simple. Le niveau d'intérêt, d'inattendu ou d'humour joue aussi un rôle pour déterminer la facilité à apprendre, ainsi que le côté “buzz” d'un item.

Quand les chercheurs étudient des items culturels complexes comme des histoires, des arguments ou même des théories ou des religions, ils sont forcés de mesurer la complexité et la facilité à être appris en utilisant des outils

plus ou moins bricolés. Habituellement, les chercheurs se servent du nombre d'éléments dans une histoire, ainsi que leurs interrelations, pour doser sa complexité. Le nombre d'éléments et de relations que quelqu'un retient couramment, est un index de facilité-à-être-appris.

Le paradigme de l'apprentissage itératif est une façon intéressante d'étudier la transmission culturelle et d'explorer comment celle-ci produit des transformations structurelles des items à transmettre. Dans le mode d'apprentissage itératif, un premier participant doit apprendre un message (une histoire, un dessin, un pseudo-langage, etc.) pour ensuite le reproduire ou le décrire à un second participant. Le second participant va alors décrire le message à un troisième, etc. Une chaîne de transmission peut comporter des dizaines de participants.

C'est certain, la complexité algorithmique n'est pas toujours utilisable dans de telles situations. Cependant, dans la recherche fondamentale en sciences cognitives, elle est appropriée. Au lieu de faire des recherches sur des éléments culturels complexes comme les rumeurs ou les pseudo-sciences, on se tourne vers des items plus simples, en espérant arriver à une meilleure compréhension de la façon dont une structure émerge dans le langage. On a utilisé certains pseudo-langages [Kirby and Paris, 1982] et montré une baisse de la complexité intuitive le long des chaînes de transmission.

Dans une étude plus récente, des chercheurs se sont servis de motifs de points en deux dimensions. Douze jetons sont placés aléatoirement sur des grilles 10 x 10. Le premier participant doit mémoriser le motif en 10 secondes, puis le reproduire sur une grille vide en utilisant de nouveaux jetons. Le second participant doit alors reproduire la réponse du premier participant, et ainsi de suite. Les participants d'une chaîne étaient tous des adultes, et parfois tous des enfants. La facilité-à-être-appris peut se définir comme le nombre de jetons correctement placés. Comme on pouvait s'y attendre, la facilité-à-être-appris augmente de manière continue le long de chaque chaîne de transmission, au même rythme pour les enfants et les adultes. Comme conséquence de cette augmentation de la faculté-à-être-appris, chaque chaîne converge vers un motif très facile à apprendre (qui est simple, aussi), différent d'une chaîne à l'autre.

Le fait que les chaînes des enfants et des adultes partagent la même vitesse d'augmentation de la facilité-à-être appris, est frappant, car ils ont des capacités cognitives différentes ..., mais il est peut-être possible de l'expliquer par la complexité. En effet, la complexité algorithmique des motifs à l'intérieur de chaque chaîne baisse continuellement, mais à rythme plus rapide dans les chaînes des enfants que dans les chaînes des adultes.

L'effet le-Moins-est-le-Plus [Kersten and Earles, 2001] en psycholinguistique désigne le fait que les bébés, dotés de faibles capacités cognitives, sont bien meilleurs que les adultes pour apprendre un langage. Ainsi, moins de capacité cognitive pourrait être concrètement plus favorable quand il s'agit d'apprendre une langue. Une explication possible de ce paradoxe est liée à la complexité : à cause du pouvoir cognitif réduit des bébés, ceux-ci pourraient être obligés de faire plus de “compression de données” – de construire une pensée plus simple, avec la possibilité d'une représentation erronée des données auxquelles ils ont accès. L'étude récente de Kempe et al. (2015) [Kempe et al., 2015] soutient cette idée (quoique de façon éloignée), en montrant que les enfants réalisent la même augmentation de facilité-à-être-appris que les adultes grâce à une baisse beaucoup plus rapide de la complexité algorithmique, ce qui pourrait être une conséquence de leur pouvoir cognitif réduit.

9.4 Remarques de conclusion

On a passé en revue la plupart de ce qui a été fait formellement pour connecter et expliquer la cognition avec la computation. Turing faisait le lien entre l'intelligence humaine et la computation. Searle a indirectement connecté la compréhension (et la conscience) à la complexité des programmes (soft IA). Tononi et al. ont récemment fait le lien entre la conscience et la théorie de l'information, et nous avons connecté cognition et théorie algorithmique de l'information. En sciences cognitives, l'étude de la mémoire de travail, du raisonnement probabiliste, et l'émergence de structures dans le langage, sont des recherches de pointe. Dans ces trois domaines, nous avons fait des contributions à l'aide de la complexité algorithmique, qui apparaît être un outil normatif très utile, éclairant plusieurs processus cognitifs. On a proposé que l'hypothèse d'un nombre fini de phrases compréhensibles est elle-même discutable : on pourrait imaginer des phrases de structure simple, cumulative, qu'on pourrait allonger indéfiniment sans qu'elles cessent d'être compréhensibles. Mais, la compréhension fait appel vraisemblablement à des procédés abstraits comme la montée sémantique ou métalinguistique qui sont de formes de compression de régularités simples.

Nous avons donc procédé en traitant l'esprit comme une sorte de compresseur algorithmique, pour un éventail de raisons, allant du fait que l'esprit opère autrement qu'en consultant des tables d'implémentations, pour aller jusqu'aux façons dont se manifestent les biais, tout ceci grâce à des moyens algorithmiques purement basés sur la théorie de la calculabilité algorithmique.

Cette recherche génère et va générer surement encore plus d'éclairages sur les fondamentaux de la cognition animale et humaine, avec une fertilisation croisée de l'artificiel vers le naturel et vice-versa, car les mesures algorithmiques des propriétés caractérisant le comportement humain et animal peuvent être utilisées dans des systèmes artificiels comme la robotique [Zenil, pear]. On dispose de ce qui équivaut à une sorte de test de Turing inversé comme décrit dans [Zenil, 2014b] et [Zenil, 2014a], qui peut servir, comme c'est proposé dans ce chapitre, contre des affirmations telles que "le web s'est développé comme un esprit global", en s'appuyant pour cela sur le fait que malgré son haut potentiel d'information intégrée, le web reste équivalent à une implémentation de type table de consultation, quoiqu'énorme, mais donc vulnérable au CRA de Searle (CRA : argument de la chambre chinoise).

Nous pensons que l'information intégrée est donc une condition nécessaire mais pas suffisante pour la conscience ; et nous avons proposé que l'affichage de capacités algorithmiques de compression (au-delà de la seule compression statistique comme implémentée dans les algorithmes classiques de compression sans perte de type entropique) soit un test de présence de compréhension, comme Chaitin l'a proposé. Cela va de pair parfois avec de l'information intégrée par le biais du concept d'expérience interne, mais pas toujours, et c'est une autre condition nécessaire pour éviter de subir le CRA de Searle s'appuyant sur des programmes de type consultation.

Pour mettre en avant cette idée, nous avons fourni des outils spécifiques dotés d'un large éventail d'applications [Zenil and Villarreal-Zapata, 2013, Zenil, 2010], valables tout autant pour le comportement animal qu'humain [Zenil, 2013a, Gauvrit et al., 2014c, Gauvrit et al., 2014a, Gauvrit et al., 2014b, Kempe et al., 2015, Mathy et al., 2014].

Chapitre 10

Conclusion générale

Dans cette thèse, nous avons d'abord exploré certaines propriétés fondamentales des mesures de complexité algorithmique, en particulier leur puissance qui est aussi la cause de leur principal défaut, celui de leur non-calculabilité. Cependant, on voit dans ce contact entre la calculabilité et la non-calculabilité qu'une compréhension profonde des aspects de la complexité d'un objet est réalisée et que des outils sont conçus pour la calculer. Certaines de ces mesures permettent des approximations finies supérieures ou inférieures, ce qui nous permet d'essayer d'approcher le non-calculable. Nous avons vu comment ces mesures ne sont significatives que sur la base des théorèmes d'invariance qui les soutiennent.

Nous avons également vu comment certaines décisions arbitraires, conventionnelles ou subjectives doivent être réalisés afin de déterminer le degré d'aléatoire d'un objet, mais (en raison de ce même théorème d'invariance) les mesures sont robustes asymptotiquement et les décisions deviennent asymptotiquement moins arbitraires ou plus pertinentes et donc plus objectives. Le fait que ces mesures soient le résultat de différentes définitions convergentes, montre aussi la puissance de cette approche, malgré le qualificatif trompeur de non-calculable.

Par contre, on a vu qu'on peut exploiter la dépendance du langage à partir des théorèmes d'invariance afin de donner un sens subjectif nécessaire pour une approche du problème du sens/signifiant, parce que les objets peuvent avoir des propriétés différentes selon des taux différents dans leur processus de convergence vers la même valeur de complexité garantie par les théorèmes d'invariance.

D'un autre côté, on a réalisé une expérimentation exhaustive pour trouver des raccourcis possibles pour distancer les computations de toutes les

machines de Turing à 3 états et 2 symboles, et de quelques-unes à 4 états, ceci en prédisant les valeurs des séries de résultats des machines pour le même nombre d'entrées. Même si cette approche ne peut fatalement pas réussir complètement à cause du problème de l'arrêt, la proportion concrète des prévisions correctes et la vitesse à laquelle ces prévisions furent prises ont été dignes d'être étudiées et mises en relation avec le concept d'irréductibilité computationnelle. Nous avons découvert que, malgré le fait que des séquences soient restées parfois incomplètes dans le cadre strict de la tentative de court-circuiter la computation des machines de Turing, aucune de ces séquences n'a finalement résisté en restant partiellement complétée. Le processus d'achèvement des séries de résultats, de temps d'activité et d'usages de l'espace des machines de Turing nous a donné aussi l'opportunité de débattre sur d'intéressants aspects de la théorie de la computation par rapport à une computation empirique plutôt que simplement théorique, incluant les vitesses de convergence, le concept d'ensembles décidables et le problème de l'extensionnalité, qui sont liés aux concepts d'irréductibilité, d'inférence inductive et d'imprévisibilité, dans les systèmes déterministes. Nous espérons que cette approche va stimuler de futurs débats et plus d'expérimentations à la fois de nature algorithmique et épistémologique.

Parce que l'une des thèses les plus importantes de ce travail est de montrer qu'il est possible d'utiliser la théorie de la complexité algorithmique pour expliquer ou donner un certain sens à des phénomènes naturels organisés et même à des phénomènes cognitifs, nous avons vu aussi qu'on est envahi par des phénomènes d'irréductibilité intrinsèque aux processus algorithmiques (et donc déterministes) et que l'impression d'être dans un monde où tout sera "mathématiquement" résolu (par exemple par la théorie du tout ou par une prétentieuse grande théorie unifiée) n'est qu'une illusion (voir aussi les travaux à ce sujet de M. Zwirn [Zwirn, 2000] et C. Imbert [Imbert, 2008]). En effet, il se pourrait qu'il y ait peu de raccourcis de calcul, vu la façon dont s'impose par exemple de manière extrême l'indécidabilité du problème de l'arrêt. Alors, au lieu de penser mathématiquement résolu, ou résoluble, ou raccourci, ou prévision, quand on pense mathématique, on devrait plutôt penser "mathématiquement suivi pas à pas", ou énumérable et non "automatiquement réductible" car, par exemple, au niveau de la complexité algorithmique, il ne peut pas y avoir autant de programmes courts que de longs, ce qui impose déjà, après le problème de l'arrêt, un autre niveau d'irréductibilité. Il est fortement probable qu'il y ait pléthore de longs programmes sans raccourcis et peu de zones de réductibilité, soit en principe soit en pratique (telles que les formules BBP qui permettent de calculer n'importe quel chiffre du développement en base puissance de 2 de la constante mathématique π

sans avoir besoin de calculer les digits précédents). Mais les problèmes d'ir-réductibilité montrent juste au départ de cette thèse que la propriété d'être "algorithmique" et même "calculable" n'implique pas "réductible", et cela est déjà fondamental pour ouvrir un dialogue qui autrement suppose qu'un monde calculable ou algorithmique voire déterministe implique automatiquement un monde réductible voire banal et donc évidemment incapable de modéliser ou d'expliquer la nature complexe du monde que cette thèse suggère.

Parmi ce que la théorie de la calculabilité de l'information algorithmique nous révèle, il y a l'omniprésence du pseudo-aléatoire si on en parle selon le point de vue de la mécanique classique sauf pour les phénomènes quantiques où un possible "véritable" aléatoire, par exemple au moment de la décohérence ou de l'émission radioactive, pourrait exister. Il y a aussi la (non-)calculabilité, qui montre qu'un phénomène peut être déterministe et imprévisible à la fois. En effet, la mécanique classique suggère que toute loi physique est déterministe, des sortes de programmes informatiques. Si c'est le cas, alors les lois physiques comme les programmes informatiques sont soumis aux mêmes lois, peut-être même plus fondamentales, des lois algorithmiques et informatiques qui déterminent ce que les lois peuvent produire et la manière selon laquelle elles pourraient distribuer des régularités à partir d'entrées simples ou même aléatoires où la distribution de Levin opérerait. Ces lois physiques pourraient donc fonctionner comme des sortes de règles de production de structure dans les trois sens de Kolmogorov-Chaitin, Solomonoff-Levin et Bennett et nous avons vu la conséquence de la puissance même des programmes simples, et leur puissance à générer des structures élaborées. En biologie par exemple, les gènes peuvent se voir comme des programmes, ou des modules fonctionnels tel que les gènes "homeobox", des programmes de programmes. On le sait parce qu'une mutation des homeobox peut faire apparaître les ailes des insectes dans une autre partie de leur anatomie, précisément parce que celles-ci sont modulaires.

Par contre, il n'est pas toujours facile de comparer des modèles de calcul abstraits avec le monde réel pour essayer de montrer qu'il leur ressemble, parce qu'on peut toujours choisir de regarder vers les parties aléatoires d'un système réel, par exemple vers la radiation de fond de l'univers. Par contre, il faut faire un choix de jugement sur la nature la plus importante du monde : est-ce que l'aspect le plus pertinent du monde est qu'il semble parfois aléatoire ou a des parties aléatoires (encore, par exemple, le bruit de fond) ou est-ce plutôt la structure, parmi elle la vie et l'intelligence humaine, ce qui est le phénomène et la caractéristique la plus remarquable de l'univers ?

Dans cette thèse on utilise la théorie de l'information algorithmique pour

essayer de donner sens ou d'expliquer la manière dont structure, sens et signification émergent d'un univers algorithmique pour n'importe quelles conditions initiales (et donc une réponse possible au paradoxe de l' "Ajustement fin de l'univers"). Même s'il y a une possibilité qu'on soit dans une petite partie ordonnée d'un univers globalement aléatoire, c'est un sujet discuté depuis plusieurs décennies dans le contexte de ce qu'on appelle les "Boltzmann brains" (Andreas Albrecht et Lorenzo Sorbo¹) qui va au-delà de la portée de cette thèse. Donc chercher des règles, c'est ce que la science fait et le pari de vouloir les extrapoler à tout l'univers peut paraître présomptueux mais supposer que la propriété la plus importante du monde soit les phénomènes aléatoires (encore, par exemple, le bruit de fond dans l'univers) est généralement rejeté ou écarté même si quelques auteurs la soutient toujours (Calude [Calude et al., 2012], McAllister [McAllister, 2003]). *Le cerveau de Boltzmann* est l'hypothèse d'un être conscient qui existe grâce à des fluctuations aléatoires. Boltzmann a suggéré la possibilité que l'univers soit observé dans un état peu probable et hors équilibre, parce que seule l'existence aléatoire de tels états permet l'existence de cerveaux qui soient conscients de l'univers. Le concept du cerveau de Boltzmann est souvent énoncé comme paradoxe physique car si l'on considère notre situation actuelle comme entités conscientes dans un monde organisé, contre la probabilité des entités conscientes dans une soupe thermodynamique sans traits bien marqués, le deuxième cas devrait être bien plus probable que le premier. Par contre, cela suppose qu'on soit dans une situation neutre et non pas dans la situation où on est, c'est-à-dire, un monde qui semble organisé. La plupart des chercheurs considèrent donc la position des "Boltzmann brains" improbable ou scientifiquement et philosophiquement improductive, semblable à un solipsisme thermodynamique où nous ne serions que des fluctuations aléatoires où on est seul et condamné à ne pas pouvoir démontrer le contraire. Dans cette thèse nous avons donc pris le choix de ne pas regarder vers la partie aléatoire de l'univers mais vers la nature organisée du monde, ce qui nous semble plus riche et important en particulier pour l'existence de tout, y compris la vie et l'intelligence.

Comme on l'a vu, au cœur de la thèse se trouve une mesure de grande élégance car elle formalise de manière mathématique deux des plus importants paradigmes en science, le rasoir d'Occam (favoriser les explications/programmes les plus simples) et le principe Epicurien (considérer toutes les hypothèses possibles, dans ce cas les programmes possibles qui produisent le même résultat) et elle montre aussi une puissance surprenante (Levin) car cette mesure domine n'importe quelle autre mesure similaire, ce qui en fait donc une mesure unique et universelle. Si comme la mécanique classique

le détermine, l'univers macroscopique est déterministe, les lois de la nature peuvent être considérées comme des programmes informatiques qui doivent être sujets à “la loi de Solomonoff-Levin” qui produit et favorise la distribution des sorties des lois d’une manière particulière même en absence de structure initiale. En effet, à partir d’un ensemble complètement symétrique (toutes les lois possibles, ou tous les programmes informatiques possibles) et de l’absence de conditions initiales ou de conditions initiales aléatoires on obtient une distribution largement (exponentiellement) asymétrique qui s’approche de la distribution universelle de Levin sans aucune autre supposition initiale (sauf la nature déterministe de l’univers à cause des lois physiques). Cette vision est très puissante et semble être testable si on compare ce qu’on voit dans le monde avec ce qu’on obtient en le simulant de manière très basique avec des programmes informatiques simples comme je l’ai fait dans [Zenil, 2013b] comme base pour cette thèse ou je développe plus cette idée.

La probabilité algorithmique de Solomonoff et Levin nous dit que la probabilité de production de certaines suites possède une probabilité d’apparition qui nous dit et nous donne les moyens de connaître sa complexité algorithmique, et vice-versa si un résultat possède une grande ou une petite complexité algorithmique, on peut estimer sa probabilité d’apparition. Au sujet de la signification, si un message est significatif parce qu’il a une probabilité faible (par exemple au sens de Shannon), il est alors d’une forte complexité, ce qui semble conforme à l’intuition. J’ai utilisé ces réflexions pour proposer un agenda afin d’essayer d’encadrer le concept de signification dans le contexte algorithmique car, par exemple, modéliser le sens/signification, permettra d’automatiser le sens. D’où un lien très fort avec la signification artificielle, ou intelligence artificielle. L’approche algorithmique n’est pas complètement étrangère aux explications moins rigoureuses mais également compliquées, par exemple Lacan (Ecrits) définit le langage humain comme une communication où l’émetteur reçoit du récepteur son propre message sous une forme inversée. Je propose que la nature subjective du message-récepteur puisse se modéliser avec le “défaut” apparent de la théorie de l’information algorithmique, celle du choix de langage de description ou du choix de la machine universelle de Turing (ce que le théorème d’invariance ne couvre pas car il ne dit rien sur la vitesse de convergence en valeurs). En donnant une piste pour modéliser la subjectivité, on peut penser que l’AIT, la complexité algorithmique, ouvre une possibilité de réfléchir avec des concepts nouveaux à une propriété essentielle de l’information (et que évidemment la théorie de la communication de Shannon ne peut pas modéliser ou expliquer). Par exemple, on a proposé qu’un modèle de la signification devrait utiliser un

concept de calcul profond pour formaliser le concept de message riche de sens, bien au-delà de ce que la théorie de la communication de Shannon toute seule pouvait nous dire par rapport au message et aux bits nécessaires pour l'encoder.

J'ai argumenté aussi pour soutenir qu'un ordinateur sans limites de ressources et réussissant le test de Turing n'est pas "intelligent, doué de conscience ou conscient de lui-même". Par contre un programme aux ressources limitées, qui réussit le test, a des chances de l'être. J'en ai conclu que le cerveau humain est certainement limité et donc il doit être plutôt algorithmique pour réussir le test de Turing lui-même. L'intelligence serait donc un raccourci de calcul. La conscience considérée comme un hyper-calcul. Il y a un temps défini quand on passe un examen, et si on avait un temps infini, n'importe qui réussirait et l'examen perdrait de sa valeur de mesure. Un caillou réussirait : il suffirait d'attendre qu'il soit transformé en ordinateur au hasard du destin industriel de sa matière première, ou qu'il soit incorporé dans un corps d'étudiant sous forme d'oligoéléments ou autre.

Quatrième partie

Appendice

Annexe A

Groupe de discussion : L'univers est-il aléatoire ?

Group de discussion, NKS Science Conference 2007
Université du Vermont, Burlington, U.S.
15 Juillet 2007

Participants : Cristian S. Calude, Gregory J. Chaitin , Paul C.
W. Davies, Karl Svozil et Stephen Wolfram

Moderator : John L. Casti

Organisateur : Hector Zenil

Transcription : Adrian German

Traduction : Bernard François

John Casti (en tant que modérateur) : Il y a quelques semaines, après avoir été contacté pour ce job par Hector (Zenil) et avoir accepté tout d'abord de veiller sur cette session, j'ai eu quelques interactions avec Stephen (Wolfram), et d'autres – sur ce dont on allait parler et comment. Et je pense que le consensus qui en est sorti fut que ce devait être quelque chose d'aussi vague que possible, mais pas plus. Je crois que tous ici ont des idées bien précises sur la plupart des problèmes dont on veut parler, certainement sur l'aléatoire, la complexité, la physique, les mathématiques, etc., Stephen (Wolfram) a émis une idée à un certain moment, qu'il a formulée comme une question quand il se demande : “L'Univers est-il aléatoire ?” J'ai trouvé que c'était une question plutôt intéressante, bien que ce ne soit pas la façon dont

je l'aurais formulée moi-même, et pour cette raison, je vais essayer un peu plus tard, d'affiner cela un petit peu si possible.

i Complexité versus aléatoire

John Casti : maintenant je voudrais focaliser concrètement non sur l'aléatoire mais sur la complexité, qui est en fait mon credo, car je pense que la complexité est une notion plus profonde que le concept d'aléatoire. Et je considère vraiment l'aléatoire comme un cas dégénéré de complexité, en un sens, car je le vois comme une notion d'extrême complexité. Et juste pour donner un modeste point de départ sur la complexité, je vais commencer en reformulant la question de Stephen comme suit : "L'Univers (ou le monde réel) est-il trop complexe pour nous ?" Et pour tirer du sens de cette question, je pense qu'on doit la déconstruire et peser tous ses mots, en lui permettant d'être assez vague pour pouvoir l'interpréter à loisir. Et pour aller plus loin là-dessus, je vais vous donner une petite liste que je tiens de Seth Lloyd. Des années en arrière, quand je suis venu pour la première fois au Santa Fe Institute, avant que Seth Lloyd devienne "Seth Lloyd", il y était comme post-doctorant, et pour le plaisir juste pour illustrer le point de vue que la complexité est tout autant dans l'il du regardeur qu'elle est inhérente à l'objet lui-même, il a écrit un article (je doute qu'il fût jamais publié) reproduisant l'histoire de Baskin-Robbins sur les 31 parfums de glace. Ça s'appelait : "Trente-et-une saveurs de la complexité". Et il identifiait 31 différentes notions et définitions que les gens ont utilisées dans la littérature pour faire référence à la complexité. Laissez-moi parcourir rapidement cette liste, je ne vais pas commenter chaque terme un par un, je veux juste les énumérer pour vous. C'est donc un ensemble, en relation avec la difficulté de description, et on y trouve la complexité algorithmique proche de la tête de liste ! Cet endroit est occupé par une autre catégorie, liée à la difficulté de création – puis on a, listées, plusieurs autres notions de complexité.

Je ne pense pas qu'il y ait quelqu'un dans cette salle, pas moi en tout cas, qui puisse réellement vous dire ce que chacune de ces choses sont vraiment – sans aller voir dans la littérature pour les y chercher. Mais ce n'est pas très important, l'essentiel est qu'il y a quelqu'un en dehors d'ici qui réfléchit sérieusement sur la complexité jusqu'à produire cette liste pour prouver qu'il existe tant de concepts différents pouvant être pris comme notion significative de la complexité. On trouve aussi le degré d'organisation, des choses comme l'excès d'entropie, la sophistication, l'entropie métrique et ainsi de suite. Et cette liste va même un peu plus loin : complexité idéale, complexité

hiérarchique, corrélation, information stockée, information algorithmique mutuelle, etc. Maintenant, si on s'étend sur les à-côtés comme je l'ai fait on ne trouverait pas seulement 31 parfums de complexité, mais plus de 40 saveurs supplémentaires à ajouter, et cela pour prouver que ce que nous voulons dire par complexité est vraiment quelque chose qui dépend du sujet abordé, en quelque sorte (c'est mon opinion).

Et si on pose la question : "Le monde réel est-il trop complexe pour nous ?" alors on doit soit ajouter quelque chose à cette liste, en partie pour répondre à la question, soit se servir de cette liste pour pouvoir y trouver ce qui puisse sembler saisir l'essence de ce qu'on entend par complexité dans notre situation particulière. Maintenant laissez-moi vous présenter quelques exemples artistiques : le premier vous montre un dessin que Karl Sims m'a donné et au-dessous vous pouvez voir le code Lisp pour ce dessin : si vous avez Lisp sur votre ordinateur et que vous lancez ce code, vous obtenez ce qu'on voit à l'écran. C'est tout-à-fait dans l'esprit de l'idée de Gregory Chaitin du "plus court programme possible pouvant servir à décrire un objet particulier". On peut compter le nombre de caractères dans ce code Lisp mais il ne faudra pas le considérer comme le plus court programme – ce qui est généralement improuvable, mais c'en est sûrement une bonne approximation – toutefois on s'en servira comme description algorithmique de cet objet-là. En dessous, on voit quelque chose qui ressemble à un mélange aléatoire de points, mais en fait il y a un motif dans cette image. C'est un exemple d'une chose très populaire, cela s'appelle la stéréographie (ou stéréocopie) et si vous arrivez à dérégler la mise au point de votre il tout en fixant l'image assez longtemps, un objet tri-dimensionnel va apparaître. On ne le voit pas maintenant que je le projette devant vous car cette image est probablement trop petite, mais l'idée est qu'elle illustre un genre différent de complexité, qui a à voir avec la notion à laquelle j'adhère personnellement comme étant l'ingrédient le plus essentiel dans un système adaptatif complexe, qui est l'idée de phénomène émergent : que les objets constituant le système interagissent de façon à créer quelque chose qui n'existe pas dans les objets eux-mêmes, initialement.

Maintenant voici un autre exemple, et je vais l'introduire sous forme de question : lequel des deux est le plus complexe ? Est-ce cet homme sur la gauche (ceux d'entre vous qui s'y connaissent en art vont reconnaître ce visage, c'est celui d'un célèbre artiste britannique nommé Damien Hurst), ou le crâne présenté à droite qui est aujourd'hui le crâne le plus cher de l'histoire, peut-être. Vous pouvez vous procurer cet objet pour une petite centaine de millions de dollars, donc qu'en pensez-vous ? Pourquoi cette chose coûte-t-elle si cher ? Qu'est-ce que c'est au juste ? Ce n'est qu'un crâne ! Mais l'artiste Damien Hurst, qui peut même prétendre que c'est secondaire, l'a recouvert de

platine et l'a incrusté de diamants pour le mettre dans sa galerie à Londres, et a dit que pour cent millions de dollars, il est à vous. Maintenant, comme point de départ, tout ce que je veux ici, c'est un exemple de système du monde réel, deux systèmes en fait – lequel des deux pensez-vous être le plus complexe ? : l'artiste ou la création ? Si vous pouvez appeler ça une création – bien que c'en soit une, mais vraiment coûteuse. Mais possède-t-il vraiment quelque chose de complexe concrètement ? C'est une question ouverte, je pense, mais intéressante.

Bon, juste pour terminer cette petite partie, mon credo, une notion de complexité que j'aime beaucoup, et qui est relativement récente, c'est l'idée que la complexité n'est pas inhérente à l'objet lui-même – que ce soit l'univers ou autre chose – au contraire, c'est très lié à l'observateur. La complexité est un concept qui doit être en interaction avec l'observateur. Pour vous donner un exemple trivial, pensez à un caillou sur la chaussée. Vous arrivez, vous prenez cette pierre et en la regardant, vous vous interrogez sur son degré de complexité. Bien, si vous êtes une personne ordinaire comme moi, sans compétence particulière [en géologie], vous allez probablement dire que la pierre n'est pas très complexe, car vous ou moi, n'avons qu'un nombre limité de façons d'interagir avec elle, en tant que non-spécialistes. D'un autre côté, si vous étiez géologue, vous disposeriez de tout un panel de façons supplémentaires pour interagir avec la pierre, qui me sont inapplicables, et la pierre deviendrait alors beaucoup plus complexe pour un géologue que pour quelqu'un d'incompétent n'ayant pas cette puissance d'observation.

Donc la complexité dans cette définition, pourrait bien être le nombre de façons inéquivalentes selon lesquelles on peut interagir avec un objet (en insistant sur le verbe pouvoir). Cela ne veut pas dire que ça doit devenir une définition de la complexité universellement acceptée ou suffisamment vague, mais c'en est une qui reconnaît au moins que la complexité n'est pas juste inhérente à l'objet lui-même. Cela amène finalement à ma version de la question de Stephen formulée comme suit : "Le monde réel (connu aussi en tant qu'Univers) est-il trop complexe pour nous ?"

Et mon argument est que cette question est au moins autant – sinon plus – une question de philosophie qu'une question de physique ou mathématiques, ainsi, partant de ce credo, je voudrais votre opinion en la matière.

ii Pouvons nous contrôler l'univers ?

Paul Davies : Bon, je pense qu'il est clair que cette introduction a été suffisamment provocante pour tout le monde ici dans cette salle et qu'il y a au moins 2-3 personnes qui veulent intervenir, d'après mon comptage.

Greg(ory) Chaitin : Je suis d'accord. D'après moi, la question est : si l'Univers est trop compliqué pour nous, alors évidemment on doit reconstruire l'être humain jusqu'à ce que l'Univers ne soit plus complexe pour nous !

Stephen Wolfram : Moi aussi, j'ai plusieurs choses à dire. D'abord que c'est peut-être une chance que le champ de la biologie n'ait pas besoin d'une définition de la vie, ou ne prenne pas pour a priori de départ d'en avoir une – autrement les biologistes ne seraient même pas capables de commencer à faire de la biologie ! Il y a en effet beaucoup à dire sur ces problèmes de complexité, mais je pense que c'est une bonne question que de partir de : "Est-il inévitable que l'Univers soit trop complexe pour nous au point de devoir nous reconstruire nous-mêmes ?" Car on pourrait assez facilement se demander : "Se pourrait-il qu'il y ait des situations où il est inévitable que nous ayons la même complexité que l'Univers ?"

Greg Chaitin : Alors – Est-il concevable que, si nous l'avons inventé et qu'il ait la même complexité que nous, nous ayons juste oublié que nous en sommes l'auteur ? C'est une question qui découle de vos deux questions.

Stephen Wolfram : Ah – Mais nous n'avons pas inventé l'Univers, n'est-ce pas ?

John Casti : Cette question ressemble un peu à quelque chose qu'on trouve dans la théorie des systèmes, où la complexité du contrôleur doit être au moins aussi grande que celle de l'objet à contrôler. On peut aussi raisonner en disant "Voyons, quelle que soit notre complexité en tant qu'êtres humains quelle que soit la notion de complexité qu'on décide d'employer, on ne pourra avoir accès qu'à la partie de l'Univers qui est aussi (ou moins) complexe que nous." Ce serait la correspondance biologique de la définition de la complexité basée sur le nombre de façons uniques d'interagir. Mais qu'en est-il si l'Univers possède un grand nombre de degrés de complexité, et nous un seul ?

Paul Davies : Donc, cela signifie-t-il, si on pose la question plus largement, que nous ne serons jamais capables de contrôler tout l'univers ? Qu'il existe une limite à la Technologie ? Si on imagine qu'il n'y a pas de limite à l'argent – ou au temps devant nous – y a-t-il néanmoins une limite à ce qui est technologiquement faisable, et nous pourrions ne jamais conquérir le Cosmos ? Est-ce cela que vous demandez ?

Stephen Wolfram : J'ai quelque chose à dire là-dessus. Un jour, peut-être bientôt, il se peut que nous ayons une théorie fondamentale de la physique. On pourrait tout savoir sur la construction de l'Univers. Donc la seule question est : une fois arrivé là, qu'en faire ? Où cela nous mène-t-il ? Alors une question évidente est : la connaissance des lois fondamentales de l'Univers pourra-t-elle nous permettre de juger si tel ou tel problème est solvable ? Ce n'est pas évident, mais dépend du problème. Toutefois, il est généralement indécidable de savoir si une solution est possible pour les problèmes donnés comme difficiles même en connaissant toutes les lois fondamentales. Ainsi, cette question est importante car nous savons qu'il ne suffit pas de connaître les lois de la physique pour être capable de mieux prévoir la nature.

Paul Davies : Au royaume des questions sur la complexité, et étant donné que John dit qu'il existe plusieurs degrés de complexité, et si on était capable de manipuler la complexité de l'inconnu, alors ma question serait : existe-t-il une sorte de limite sur l'Univers ? Je parle d'une limite de complexité qui fonctionne sur notre Univers, disant qu'on ne peut atteindre qu'un certain niveau de contrôle sur l'Univers – au-delà, il y aurait une certaine limite insurmontable sur la capacité, à laquelle nous aurions à faire face.

Stephen Wolfram : Peut-être quelque chose comme le second principe de la thermodynamique ?

Paul Davies : Oui, comme le second principe de la thermodynamique mais pas tout-à-fait pareil, voyez-vous, en faisant certaines choses très habilement – et c'est ce qui m'intéresse vraiment – ce n'est pas une quelconque sorte d'information, ce sont des sortes d'information très spécifiques qui nous permettent de manipuler notre environnement. Donc, si nous ne comprenons pas les mouvements des astéroïdes dans le système solaire, nous pouvons toujours provoquer certains changements, par ex. avec une petite poussée sur un astéroïde, nous pouvons le faire heurter une planète, qui devient une planète avec une nouvelle trajectoire, et par une accumulation suffisamment habile de ces changements, je pense que nous pourrions finir par configurer nos galaxies. Mais il doit y avoir une limite à ce qu'on peut faire ! Nous ne serions pas capables de travailler sur l'Univers et encore moins capables de vraiment l'expliquer, ne pensez-vous pas ?

Stephen Wolfram : Mais voyez-vous, il se peut qu'il y ait deux problèmes : l'un d'eux est qu'il puisse y avoir de l'indécidabilité et/ou de l'irréductibilité pour trouver ce qui arriverait ...

Paul Davies : Aucune d'elles ne serait indésirable !

Stephen Wolfram : D'accord. Alors la question est de savoir s'il existe

un moyen de configurer les astéroïdes pour envoyer un grand message dans l'Univers.

Greg Chaitin : Disons qu'on serait dans un Univers qui est un morceau de logiciel tournant sur un ordinateur ... Il se pourrait que le logiciel ait une routine quelque part où il se réécrit lui-même. Bon alors, il se pourrait que si on comprend l'Univers suffisamment bien, on puisse changer les lois de la physique, ou les modifier dans le futur comme par l'intermédiaire d'un morceau de code auto-modificateur !

Stephen Wolfram : Si je peux faire une interruption – apparemment, nous sommes tous d'accord sur le fait que l'Univers est vécu comme une émulation de la réalité. Donc vraisemblablement, entre les lois fondamentales de la physique et ce dont on a concrètement l'expérience, il y a beaucoup de couches intercalées car nous ne vivons pas les choses à un niveau aussi bas que les premières couches, on a l'expérience d'une certaine théorie effective de ce qui arrive. Donc peut-être dites-vous qu'on pourrait réécrire les lois au plus bas niveau du système d'exploitation, de sorte que ce que l'on vit soit comme dans le film “Matrix”, quelque chose de très différent des lois courantes de la physique et ainsi la question est – étant donné les lois fondamentales de la physique, existe-t-il un moyen d'assembler la matière dans l'univers de façon à ce que le comportement effectif de l'Univers soit tout-à-fait différent du comportement que nous avons l'habitude de vivre ?

Greg Chaitin : Mais avec cette conception des choses, comment tu définirais un bug ? Car on pourrait crasher l'Univers en attrapant le code qui va dans le mauvais sens. En général, tu désires quelque chose de constructif, mais ça peut virer à la destruction, et ainsi ça va devenir plus dur de dire ce qui se passe.

Stephen Wolfram : Bon, par exemple ça pourrait être un nouvel état de l'Univers. Pour prendre une possibilité très simple, quand j'étais plus jeune, on avait l'habitude de dire, au sujet des noyaux, que les pions soustrayaient des nucléons, que les mésons bruts les repoussaient, peut-être que les [demi-]mésons en rajoutaient ... Globalement, cette idée s'est avérée ne pas avoir de sens, mais l'important est que si on pouvait suffisamment écraser un noyau, il deviendrait soudain un autre type de noyau. Peut-être qu'il existe un autre état de l'Univers qui n'a jamais été déclenché.

iii Esprits, observateurs et l'univers

Paul Davies : Il semble cependant qu'il pourrait bien y avoir un esprit, des esprits dans l'espace extérieur qui soient d'autres types d'esprit que le nôtre, d'autres types d'êtres vivants que nous, habitant le même Univers et pour lesquels la complexité serait différente de celle que nous percevons. Car lorsqu'on pense complexité, on a toujours ce préjugé d'accepter et d'assumer seulement notre position en pensant qu'on est les seuls dans les environs. Mais il est possible d'imaginer que la complexité de l'espace extérieur vu par les yeux d'autres êtres vivants soit différente, bien que ce soit le même monde, mais vu sous une autre perspective – peut-être computationnelle ou autre chose – cela pourrait avoir un air très différent.

Stephen Wolfram : Mais il faudrait s'attendre à ce que ces "êtres" ... Je veux dire – pourquoi les appeler "êtres" ? Peut-être parce qu'on est forcé de les identifier à un certain type d'intelligence.

Paul Davies : Oui, oui ... ce sont des observateurs ...

Stephen Wolfram : Qu'est-ce qui les fait considérer comme des observateurs ?

Paul Davies : Bon, "Qu'est-ce que ça fait d'être comme ça ?" comme la fameuse question de Thomas Nagel dans son article. On peut essayer d'imaginer ce que c'est d'être comme eux. Une certaine sorte d'existence ... Qu'est-ce que c'est d'être différent de ce que nous sommes maintenant ?

Greg Chaitin : C'est ce dont on a besoin à la base pour un modèle fondamental.

Stephen Wolfram : Mais souvent les gens disent "imaginez ce que se serait d'être comme un ordinateur" ou "ou imaginez d'être comme [je sais pas quoi]" et ainsi de suite. Et essayer d'imaginer ne dit pas ce que se serait vraiment. Mon favori est : "Le temps [qu'il fait] possède son propre esprit" Mais peut-être que ça ne rentre pas dans vos critères. Difficile de réellement imaginer avec précision ce que se serait d'être quelque chose d'autre...

Paul Davies : C'est exactement ce que voulais dire.

Stephen Wolfram : ...mais tout de même, on sent que cette autre existence doit avoir quelque chose comme un esprit.

Paul Davies : Oui, mais je suis pas certain pour la météo.

John Casti : Revenons à la question du début sur la possible existence de limitations inhérentes à notre capacité de manipuler l'Univers, en quelque sorte...

Paul Davies : Je pense que ça pourrait être une bonne question de science et un étudiant en doctorat serait très content de travailler là-dessus.

John Casti : Vous pensez qu'on arriverait à quoi comme résultat de cette recherche ?

Paul Davies : Je pense qu'il y a une limite, en effet.

Greg Chaitin : On s'en moque qu'il y ait une limite?! Il s'agit juste d'imaginer que c'est...

John Casti : Non, je pense que c'est intéressant car la question n'est pas tant de savoir s'il y a effectivement une limite, ou qui s'en préoccupe. La question est " Où est la limite ?" Et "Que faire pour la repousser ?"

Karl Svozil : Je pense que ma position est ceci – Je crois vraiment que notre capacité à manipuler l'Univers est seulement limitée par notre imagination et elle seule. Je crois que quoiqu'on puisse imaginer pouvoir faire sur quoi que ce soit, on arrivera à le faire d'une manière ou d'une autre.

Stephen Wolfram : On peut penser au second principe de la thermodynamique comme simple comparaison théorique et poser la question : étant donné une certaine énergie sous forme de molécules et positions aléatoires dans le vide, pouvons-nous réussir à extraire ce qu'on veut de cette configuration aléatoire de bits ? Et je suppose que l'une des leçons de la technologie est qu'on dirait classiquement : une fois que cette chose est chauffée, tout se perd et on ne peut rien en extraire d'utile, mais si on arrive à faire plus de calculs, on va réussir à obtenir certaines approximations très bonnes et en extraire vraiment quelque chose par ce biais – bon, concrètement (parlant à Davies) vous devriez être du même avis, vu que les grandes lignes de votre premier livre vont dans ce sens ?

Paul Davies : Ok, ok, je pense – je prétends qu'il y a un certain prix thermodynamique à payer et on peut approcher certains objectifs idéalisés de ce genre, mais je ne crois pas qu'il y ait d'entropie négative en soi, car à y bien réfléchir, on arrive à faire beaucoup de choses en ayant une stratégie, on peut minimiser le coût thermodynamique pour réaliser certains de ces objectifs.

Stephen Wolfram : D'une certaine façon, l'objectif en question est de décoder l'Univers, et non d'être capable de savoir vers où le diriger.

Paul Davies : Ok ok ok – mais c'est pourquoi je pense que c'est directement lié à l'objectif d'un esprit (en tant qu'opposé à juste des intentions hautes) et c'est pourquoi je pense que les observateurs sont significatifs et non par exemple optionnels ou vus simplement comme des contraintes. Car,

en un sens, je pense qu'un observateur, ou un esprit si vous voulez, pourrait en principe déterminer la relative complexité de l'Univers, et je me demande concrètement jusqu'où, jusqu'à quelles limites.

Stephen Wolfram : Mais certains d'entre nous pensent que la plupart de ce que fait l'Univers est computationnellement semblable à ce que l'esprit fait.

Paul Davies : Je sais.

Stephen Wolfram : Donc l'esprit n'est pas au-dessus de l'Univers et parce que nous avons des esprits, voyez-vous – (il décide de reformuler :) Disons que nous sommes la première "intelligence", selon nous, ou les premières choses intelligentes survenant dans notre Univers, ainsi il se pourrait que tout le futur de l'Univers dépende grandement de cela et qu'un beau jour, voyez-vous, l'Univers ait un aspect très différent précisément à cause des esprits présents en lui. Ça se pourrait – mais je pense que c'est hautement improbable.

Paul Davies : Pensez-vous que cela se réaliserait sans la présence d'esprits, ou que ça n'a aucune chance de se réaliser ?

Stephen Wolfram : Il y a déjà tout un tas de choses qui existent, tout cela présent dans l'Univers, on regarde et on se demande : Est-ce qu'il a fallu un esprit pour fabriquer tout ça ? Quelle chose spéciale déciderait de faire un esprit ? Si on arrivait à une technologie future arbitrairement puissante – du genre qui puisse déplacer étoiles et galaxies – qu'est-ce que ferait un esprit futur avec ça ?

Paul Davies : D'accord, bon, pour poser une autre question que la vôtre, mais qui est liée, car je suis porte-parole de la détection SETI à mes heures perdues. Ainsi la question qui me vient est si j'attrape ce signal d'un E.T. pendant mon tour de veille – ça va venir d'un être doué de sensation, ou il est possible qu'on ait affaire à quelque chose ayant l'apparence de l'intelligence, comme un phénomène émergent ?

Greg Chaitin : Voyons si je peux synthétiser la position de NKS à ce sujet. La position de NKS est que tout calculateur universel est un esprit et que la computation universelle est présente partout dans l'Univers. Donc l'Univers entier possède un esprit !

Stephen Wolfram : Ou bien, d'une façon plus opérationnelle, pensez aux pulsars. Quand on a détecté des pulsars pour la première fois, on a pensé d'abord que ce signal très spécial devait être d'origine intelligente pour arriver plus tard à d'autres interprétations – et aujourd'hui nous savons que les pulsars ne sont tout bonnement pas des êtres intelligents au sens où nous

l'entendons. Maintenant si vous regardez les pulsars en détail, vous allez voir qu'ils présentent effectivement un type de complexité très intéressant et même fascinant, et j'affirme que ces séquences qu'on observe tirées des blips, tirées de la magnétosphère et autres, possèdent des bits qui font penser à un esprit comme n'importe quoi d'autre observé n'importe où, quel que soit ce qu'on décide d'observer.

iv Esprits qui communiquent

Paul Davies : Cela laisse supposer qu'on pourrait essayer de chercher un message venant des pulsars.

Stephen Wolfram : Bien – Qu'est-ce qu'un message ?

Paul Davies : Ce serait quelque chose qu'on pourrait reconnaître comme véhiculant de l'information, même si c'est à une échelle infinitésimale.

Stephen Wolfram : Mais c'est un message très inefficace !

Paul Davies : Ca se pourrait bien, mais c'est ce que nous avons coutume de faire.

John Casti : Il faut dire que votre perspective anthropomorphise à la base à outrance ...

Paul Davies : Totalement d'accord.

John Casti : Vous dites : "un message est quelque chose qu'on serait capable de reconnaître comme significatif, etc. " Et je pose en effet la même question : si on en tire du sens, on est au moins aussi complexe que ça l'est, au sens propre. Cependant dans ma définition, je n'exclue pas d'autres usages ou autres significations possibles, en disant que l'observateur, privé de la connaissance nécessaire, ou privé de la complexité nécessaire pour être relié à cela, ne pourra pourtant pas le voir. Et tout ça tourne autour de l'idée d'un esprit qui serait le support pour cette émergence. Si la complexité est l'il du regardeur, l'esprit est le regardeur et il en faut un, au moins. D'un autre côté, l'idée que tout est esprit – c'est un petit peu trop généreux, peut-être : c'est comme si toute déclaration était un théorème, ce qui ne nous aide pas trop, car on ne peut pas faire la distinction entre esprit et non-esprit, si tout a un esprit.

Stephen Wolfram : Mais il y a une nette distinction entre... (pause pendant un temps, puis reprise :) Ecoutez, la plupart de ce qu'on a étudié en physique dans le passé n'est pas semblable à un esprit, c'est en fait trop

simple pour y ressembler. L'idée que les gens ont eu depuis les temps anciens (et c'était la pensée dominante au sujet des choses et je ne pense pas que c'est faux, comme les scientifiques modernes voudraient le faire croire) était de dire qu'il existe beaucoup de choses qu'on peut prédire en se basant sur la considération que le temps qu'il fait, la météo, possède un esprit en propre – vous pouvez déterminer jusqu'où quelque chose est concevable ou prévisible et ainsi de suite. Mais par rapport à SETI, je serais vraiment curieux d'aller plus loin et découvrir si c'est un message.

Paul Davies : Ce serait bien !

Greg Chaitin : Puis-je dire ce que je pense de ce que serait le bon message à envoyer ? Ce ne serait pas un programme télé ou quelque chose comme ça. Fred Hoyle a eu une autre idée, dans un autre livre – qui était un sacré mauvais bouquin, je préfère de loin le "A pour Andromède" – l'idée de Hoyle était que le genre de message à envoyer devait être un morceau de logiciel qui cherche à se propager. Si vous envoyez un programme d'ordinateur et évidemment un manuel d'instruction pour l'ordinateur/la machine sur laquelle ça doit tourner, alors, dans cette histoire de Fred Hoyle, les gens construisent un ordinateur et font démarrer le programme sur lui, et c'est une intelligence artificielle. Ainsi, ce serait une façon de s'envoyer soi-même à travers l'Univers, pour des organismes qui seraient des logiciels.

Stephen Wolfram : C'est le ver informatique ultime.

Greg Chaitin : Oui, ce serait une forme de vie qui se transmettrait elle-même comme un message. Et il me semble que ce serait le seul message intéressant : un message qui désire venir ici et commencer à tourner ici car c'est la seule façon intéressante de s'auto-envoyer à travers l'Univers : comme message, si vous êtes un programme. Vous êtes un programme d'IA qui vient sur Terre et commence à tourner. Et du coup, vous êtes là !

Paul Davies : Cela mène à une intéressante question sur ce qui serait le moyen le plus efficace pour décoder le message.

Greg Chaitin : Mais le message est – mais le moyen de dire que c'est un message, c'est l'idée que vous faites tourner le logiciel, et s'il commence à discuter avec vous et qu'il veut renverser le gouvernement, alors c'est un message qui a du sens ... autrement il aurait l'air aléatoire ...

Stephen Wolfram : Je vois que vous voulez dire. Mais il y a un fait intéressant : avant, quand on écoutait des bandes radio, on entendait toutes sortes de choses manifestement significantes. De nos jours, avec tous les téléphones portables et toujours plus d'autres appareils digitaux, on ne peut plus

rien comprendre car tout est compressé. Et donc l'idée de signaux d'émissions radios ou télé, ça semble être de la vieille nouvelle.

Paul Davies : Peut-être que je suis en train de devenir très sceptique sur les approches les plus classiques de SETI. Je pense qu'on a besoin d'approches radicalement nouvelles pour traiter la question.

Stephen Wolfram : Mais vous pensez que ça aura quand même un sens, je veux dire que la question elle-même a un sens ? Car s'il y a de l'esprit partout, ça veut dire ...

Paul Davies : C'est ce qui me choque dans cette discussion car si vous pensez que c'est le cas, alors il y a une ou plusieurs de ces choses là-dehors et elles cherchent à communiquer ...

Stephen Wolfram : Mais je pense que le problème est juste comme l'IA. Qu'est-ce que ça signifie d'avoir une IA ?

Paul Davies : Bon, on sait ce que ça veut dire que d'avoir une conversation avec un autre être humain.

Stephen Wolfram : On sait ce que ça veut dire que d'avoir une IA de type humain.

Paul Davies : Oui.

Stephen Wolfram : C'est quelque chose qui est très proche de nous.

Greg Chaitin : Plus comme un accident historique. **Stephen Wolfram** : Aussi la question est combien est-ce qu'on peut enlever de détails de notre humanité et rester dans quelque chose qu'on sait être intelligent.

Paul Davies : Laissez-moi répondre à cette question. On peut imaginer la vie sur une autre planète, SETI classique, mais pouvons-nous imaginer récupérer un message d'un système physique qui n'est pas passé par ce processus d'émergence de la complexité organique et de l'évolution culturelle ? Sommes-nous, en quoi que ce soit, prêt à recevoir et comprendre ça ?

Stephen Wolfram : Je pense à un exemple qui peut être pertinent par rapport à ça : vous connaissez le chant des baleines, n'est-ce-pas ? Les chants des baleines servent vraisemblablement à communiquer quelque chose qui a du sens. Vraisemblablement – car ce n'est pas clair, je ne pense pas que quelqu'un en soit complètement sûr de toute façon. Je pense que personne ne sait s'il y a corrélation entre ce que dit ou chante une baleine, et ce qu'elle fait. Disons que c'est un message. Quand nous écoutons les chants des baleines, c'est vraiment difficile de dire de quoi ça parle. Ici arrive une anecdote historique amusante : il y a déjà un moment, quand la radio en était à ses débuts, Marconi, qui fut l'un des développeurs de la radio, possédait un

yacht et il traversait l'Atlantique avec, et au milieu de l'océan il écoutait avec son antenne radio, il a entendu une sorte de ... woo-eeee-ooaah. Je ne peux pas vraiment le refaire mais c'est ce que ça fait quand on écoute ces bruits – bruits qu'il a capté avec son mât radio. Bon, qu'est-ce qu'il en a conclu sur ce que c'était ? Sa meilleure hypothèse fut que c'était les martiens ... Il en a déduit que c'était des messages radio venant des martiens. Il s'est avéré que c'était des sifflements rythmés ionosphériques dûs à la magnétohydrodynamique. Donc cette histoire sur ce qu'a capté Marconi est un exemple où on entend quelque chose qui est au moins apparemment indiscernable de ce qu'on peut entendre d'un système physique de type esprit, et vraisemblablement d'un système biologique de type esprit.

John Casti : Je crois vraiment que cette histoire de baleine soulève un point très intéressant et important pour SETI, en l'occurrence : au fil des années, on a pu observer quantité d'expériences différentes – des gens essayaient de parler aux dauphins, aux chimpanzés, aux pieuvres et autres – et j'ai largement l'impression que la communication fut si limitée dans la plupart de ces expériences qu'on pourrait facilement et simplement les qualifier d'échecs, toutes autant qu'elles sont. Echec au sens où le niveau de communication réelle a été tout-à-fait minimal. On a essayé ici de communiquer avec des organismes avec lesquels on partage une très longue histoire évolutive, tant et tant de siècles, et on obtient si peu de résultats quant à établir une significative communication – si tant est qu'on croit avoir effectivement communiqué. Comment réussir à imaginer que si jamais on attrapait un signal venant d'une onde émise par une civilisation distante – avec laquelle on n'a rien en commun – qu'on serait réellement capable de comprendre le message ?

Paul Davies : Vous soulevez un problème valable, mais ...

Greg Chaitin : Puis-je renverser la question, si je peux interrompre, je pense qu'on devrait traiter ça par l'autre bout, Paul : on devrait être le SETI, on devrait être le message, donc qu'est-ce que ça serait d'envoyer un message en espérant que ceux qui le reçoivent le fasse tourner sur l'ordinateur, et que le résultat du programme soit nous-mêmes – ou une copie de nous. Ce serait un programme exploratoire IA qui ferait le tour et observerait la planète sur laquelle il est, puis nous enverrait cette information dans un message qu'on pourrait comprendre.

Paul Davies : Cela a été fait.

Greg Chaitin : Ah oui ?

Paul Davies : Oui, mais pas tout-à-fait comme ça ...

Stephen Wolfram : Hé, pas comme ça, rien à voir avec ça ! Ce qui fut

envoyé a été une chose absurde !

Greg Chaitin : Ok, pour être plus précis, laissez-moi expliquer ce qu'on enverrait : ce serait l'envoi d'exemples de listes très simples, de façon à être clair sur la sémantique de notre langage de programmation. Puis on commencerait l'envoi d'images de la liste dans le cadre d'interactions avec le monde extérieur, car j'aimerais que la liste soit associée à une caméra et ... Je sais que c'est difficile, mais ce n'est pas un programme complet, sinon je l'aurais déjà fait. Ensuite, vous envoyez ce long programme qui est en fait vous, qui commencerait à se balader et interagir avec les autochtones en essayant de trouver de l'information pour nous la retourner. Et là, on est sur place : c'est parti ! On est les SETI et l'idée est d'attirer l'attention sur nous de façon à ce qu'on soit activé. On veut envoyer un bout de code que les gens seraient curieux d'essayer de faire tourner sur les ordinateurs pour voir ce qui se passe ! C'est comme ça que ça se passerait là-bas : on se met nous-mêmes à l'intérieur de cet autre environnement ...

Stephen Wolfram : Au fait, je pense à ce truc sur la communication avec/vers les animaux et autres, mon idée personnelle à ce sujet (je pense que je dois vous avertir que je vais dire quelque chose de légèrement provocateur) mais ma pensée est que dans les prochaines années, si quelqu'un se met concrètement à travailler là-dessus avec les techniques modernes, je pense qu'il pourrait bien en tirer beaucoup plus que ce qui a été fait jusqu'ici, et voici comment : pour l'instant, (quand on essaye de parler à un chimpanzé ou un autre animal) on n'enregistre pas tout de la vie du chimpanzé, ni tous les types ou/et les façons selon lesquelles on interagit. Je parie que si c'était fait, enregistré correctement et corrélé avec les expérimentations, on pourrait effectivement commencer à produire un mécanisme raisonnable de communication. Je pense qu'il existe déjà certaines procédures prometteuses, et mon exemple favori est le (et je me suis aperçu que c'était un produit labellisé sous le nom de) Bow-Lingualg, que certains d'entre vous peuvent connaître, qui est cette chose où quand vous tapez ce que vous voulez dire en anglais, ça ressort en aboiements de chiens. Et il y a aussi quelque chose pour écouter les aboiements du chien – et vous devez l'ajuster, le régler pour le type particulier de chien que vous avez – et ça analyse le format d'après une sélection tirée d'une petite collection de types d'états, surtout émotionnels, de choses à communiquer, et ça vous donne ce que le chien disait quand il aboyait.

John Casti : Ça ressemble un peu aux fameux dessins humoristiques new-yorkaish : deux chiens assis dans une pièce, l'un deux à côté d'un terminal d'ordinateur en train de taper au clavier, et l'autre assis par terre, quand celui qui écrit dit : "Tu sais, sur internet, personne ne sait que tu es

un chien !"

v Le problème de l'Intelligence Artificielle

Stephen Wolfram : Disons que vous réussissez à établir à un certain niveau une communication avec un type quelconque d'être. Le problème est alors : de quoi vous allez parler et quels sont les motifs qui paraissent faire sens dans le contexte de cette conversation ? Vous parliez de la signification, et la signification est étroitement liée à l'intention : comment savez-vous si la chose a une intention, qu'elle a une signification et un motif ? Quels sont les prémisses et/ou les intentions de l'E.T. qu'on recherche.

Paul Davies : Je comprends votre question. Bon, la réflexion derrière le SETI classique, si on peut revenir un peu là-dessus, fait incroyablement preuve d'étroitesse d'esprit. Elle suppose en gros qu'ils sont comme nous mais qu'ils sont là depuis beaucoup plus longtemps et qu'ils ont amassé plus d'argent, et ainsi ils vont nous envoyer des messages, et c'est soi-disant clair comme de l'eau de roche qu'ils vont supposer qu'on vient juste de tomber par hasard sur la technologie radio et donc ils vont faire en sorte que ce soit très évident que c'est un message et – dit autrement, ils vont prendre les commandes !

John Casti : Agissant peut-être un peu comme ces missionnaires quand ils recrutaient les indigènes pour leur église, et on sait que cela n'a pas été une expérience particulièrement réussie, surtout pour les indigènes.

Stephen Wolfram : Oui, aussi réussi que Greg l'a dit, simplement. Mais je pense que le meilleur argument que j'ai entendu venant de la communauté SETI est : on pourrait bien faire ce qu'on peut faire.

Paul Davies : Oui, c'est vrai qu'on ne fait essentiellement qu'écouter là dehors ... Mais ce qu'on a vu et peut-être que je vais sortir des rails, mais mon point de vue sur ça est qu'il n'y a pas de raison qu'E.T. envoie des messages avant de pouvoir être sûr qu'on est là, et ils ne savent pas qu'on est là au-delà d'environ une centaine d'années-lumière de distance, et c'est plutôt improbable qu'il y ait une civilisation avancée à l'intérieur de ce périmètre. Ainsi la stratégie peut paraître sensée mais c'est peut-être quelques centaines d'années trop tôt pour la commencer.

Stephen Wolfram : Je pense honnêtement et finalement ... Je dois dire, je pense que toutes ces choses – je veux dire cette question sur ce qui compte en tant qu'esprit et quelles intentions ...

Paul Davies : Bien, c'est ce qui m'intéresse !

Stephen Wolfram : ...oui, bien – mais je pense que ce problème est profondément embrouillé. Je veux dire, j'ai été un grand enthousiaste de SETI et j'ai trouvé toutes sortes de choses sur comment on pourrait utiliser des satellites de communication commerciale pour traiter merveilleusement des données de radiotélescopes en se servant des pièces de rechange inutilisées – il y a tout un tas de grandes choses qu'on pourrait faire. Mais après avoir commencé à travailler sur NKS, je suis devenu en quelque sorte beaucoup plus pessimiste là-dessus ...

Paul Davies : Bon, mais alors vous avez dû – parce que si votre vision est que l'Univers est rempli d'esprits et peut-être d'esprits communicants, et peut-être que ça se passerait dans des systèmes aussi variés que les pulsars que vous mentionnez et autres – alors vous avez dû ...

Stephen Wolfram : Ok, mais qu'est-ce qu'un esprit communicant ? A chaque fois qu'un système physique a un effet sur un autre ... c'est un esprit communicant !

Paul Davies : C'est juste. Mais alors on aurait l'exemple d'un esprit de l'Univers plutôt appauvri et particulier, sinon on peut montrer que si on représente une classe entière, vous voyez, d'intelligence biologiquement instanciée ou quelque chose, peu importe son nom, alors il peut y avoir des super-esprits ou esprits bizarres – tout comme le nôtre – s'agitant partout aussi, donc s'il y a des gars comme nous là-bas, alors tout ça est digne d'être repris ...

Stephen Wolfram : Alors la question est quelle est la probabilité qu'il existe des esprits – bon, les esprits peuvent être courants – la question serait "Quel degré d'humanisation ont les esprits à l'intérieur de la distribution de tous les esprits ?"

Paul Davies : C'est là où ne disposer que d'un seul de ces ensembles est une terrible chose car il est très facile de penser à toutes sortes de raisons pour lesquelles ça devrait être de cette façon : ainsi, si vous projetez l'évolution ailleurs, vous vous dites "Ben oui, bien sûr, il faut qu'il y ait un traitement de l'information et des organes des sens, et alors évidemment vous vous voulez que ce truc décolle, et si ça marche vous avez besoin de quelque chose pour éviter les accidents, etc." et vous pouvez facilement poursuivre l'argument jusqu'à un type humanoïde d'entité – si on ne fait pas attention !

Stephen Wolfram : C'est pourquoi on ne peut pas comprendre quelque chose d'autre – ce qui est en soi très embarrassant !

Paul Davies : Mais je pense que, ayant suivi SETI moi-même pendant 20-30 ans, j'ai été étonné de voir comment même des changements mineurs dans la technologie se sont reflétés sur les changements de la stratégie de base de SETI. Vous rappelez-vous que dans les premiers temps, la question essentielle était quelle fréquence ils devaient utiliser ? Le problème principal était d'essayer de définir un domaine de couverture ! Personne ne parle plus de ça car on peut maintenant couvrir simultanément des milliards de canaux. La technologie a tellement évolué en à peine 10-20 ans que cela a modifié complètement la stratégie de recherche. Dans 1000 ans, on pourra peut-être dire "Evidemment, on utilise ça et ça ..."

Stephen Wolfram : Je pense que l'invariant de SETI est que leur effort de recherche est (con)centré sur les avancées technologiques terrestres les plus significatives du moment, par exemple l'histoire des canaux de Mars s'est produite juste au moment où on construisait des canaux sur Terre, et généralement il y a une grande influence de la technologie du moment sur Terre.

Greg Chaitin : Il y a une merveilleuse histoire de SF là-dessus. L'idée est que certaines personnes sont en train de faire une expérience SETI sur Terre, et c'est leur grand projet, et tout ça. En même temps, il y a des gens sur d'autres planètes, ce sont des E.T. qui essayent aussi de communiquer – et on en est là, on n'a découvert absolument rien qui utilise les ondes radio ! Dans le même temps, les rayonnements neutrinos que cette autre planète considère être le seul mode logique possible de communication bombardent des messages urgents qu'on ne peut pas voir, et les couchers de soleil sont modulés par certains autres rayons, d'une façon qu'un autre groupe d'êtres suppose être le plus évident de tous les moyens possibles de communiquer avec d'autres êtres humains rationnels (ils avaient pensé que c'étaient à l'aide de couleurs). Donc le ciel reçoit des flashes de neutrinos émis par une civilisation et aussi ce message plein de sens au soleil couchant envoyé par une autre seconde civilisation, pendant qu'on se fige dans l'écoute du spectre des fréquences radio, et personne en communique avec personne !

Stephen Wolfram : Très intéressant mais revenons à la réalité de notre discussion. L'Univers est-il aléatoire ou non ?

vi Les mathématiques sont-elles inévitables ?

Cris(tian) Calude : Puis-je ... Je pense que l'Univers est aléatoire. Et même si ce n'est pas vrai, c'est encore productif pour l'esprit humain de

croire en l'aléatoire de l'Univers, et je vais vous donner ma très courte explication de pourquoi c'est possible qu'il en soit ainsi. Toutes les lois de la nature sont des idéalizations mathématiques – essentiellement des moyens d'approcher le fini grâce à l'infini. La puissance des mathématiques en tant qu'outil de modélisation et basée sur le fait qu'elles introduisent l'infini en tant qu'estimation du très grand fini. Toutes les équations que nous utilisons en physique introduisent essentiellement l'infini, qui est très beau et parfait d'un point de vue mathématique mais qui ne peut être vérifié en réalité sauf pour de très petits échantillons. Ainsi, je pense qu'il est plus approprié de ne pas parler de l'Univers en général, mais de parler seulement de règles locales. Probablement, le plus simple exemple est de vous donner un million de digits d' Ω et de prétendre qu'on a là une loi de l'Univers – mais cet échantillon est juste une régularité locale dans l'aléatoire infini.

Stephen Wolfram : Il y a quelque chose (une partie) de ce que vous avez dit que je n'ai pas bien compris, ou peut-être c'est juste que je ne suis pas d'accord. C'est l'affirmation que la physique traite, d'une certaine façon, de l'infini. Si cela peut sembler être le cas, c'est seulement parce que la physique traditionnelle, telle qu'elle a été formulée au cours des derniers trois cents ans, s'est servi des idées du calcul – et elle a été développée autour de ces notions de calcul. Si les Automates Cellulaires s'étaient développées avant le calcul, je pense que l'histoire aurait été entièrement différente. Cela revient à dire que la physique, comme on le voit couramment, se trouve utiliser les équations différentielles mais rien en physique ne dit que c'est la réalité. Il n'y a rien en physique qui soit précisément une équation différentielle, absolument rien. Qu'on regarde la dynamique des fluides ou autre, on voit un tas de molécules rebondissant et elles se trouvent être bien approximées par Navier-Stokes. Les mouvements des planètes ne sont pas des équations différentielles, mais il se trouve que ces équations les décrivent bien. Mais il y a beaucoup de détails qui ne le sont pas. Je pense que les lois de la physique comme elles ont été formulées ces 300 dernières années en terme de mathématiques de l'infini, sont juste une coïncidence de l'histoire. Je ne pense pas que cela soit le reflet de quelque chose de profond en physique.

John Casti : Je voudrais faire un commentaire là-dessus parce que ce serait riche d'enseignement de suivre cette expérience de pensée : imaginons que Newton ait eu un super-ordinateur. De quoi aurait l'air les mathématiques aujourd'hui ? Je pense que toutes sortes de choses qui font partie des mathématiques traditionnelles, comme tout le matériel sur l'infini des suites, des limites et du calcul lui-même, ne se serait pas du tout développées, ou du moins pas au niveau où il l'est aujourd'hui, et on aurait développé à la place des théories du fini et des mathématiques du fini, différents types de

choses combinatoires en relation avec l'informatique, donc je souscris complètement à ce point de vue que c'était un accident de l'histoire. Même l'idée de continuum est simplement une question de commodité calculatoire ...

Stephen Wolfram : Je pense que c'est un point de vue controversé! Mais j'ai été fasciné par ce que d'autres ont dit sur ce sujet, ou ont pensé : "Est-ce que le continuum est vraiment un accident de l'histoire?"

John Casti : Ou une question de commodité calculatoire?

Paul Davies : Peut-être est-ce une question d'idéalisation platonique?

Greg Chaitin : Bon, il n'y a personne qui pense que la géométrie est bâtie à l'intérieur du cerveau humain, et que les yeux possèdent une façon intrinsèquement humaine de traiter les images? Que le cerveau humain a une formidable intuition pour le traitement géométrique, pour les images, et qu'une grande partie du cerveau traite tout cela grâce à un hardware dédié ...?

Stephen Wolfram : Donc vous êtes en train de dire que la raison pour laquelle on conçoit le continuum vient du fait que nos cerveaux sont construits de telle façon que c'est pratique pour nous de le concevoir. Ce qui rendrait le continuum nécessaire, dans un certain sens, de sorte qu'on ait la possibilité de concevoir le monde qui nous entoure.

Greg Chaitin : Peut-être – Je m'excuse, je vous jouais une parodie de Kant.

Cris Calude : Je suis d'accord avec ce que vous avez dit, à la fois sur le calcul comme étant un accident historique, et à la fois pour les choses que vous avez soulignées – mais je ne vois pas en quoi cela peut faire changer mon point de vue, car les AC sont tout de même utiles en ce qu'ils modélisent les propriétés de l'infini grâce au fini, et ensuite si vous réfléchissez que la plupart des propriétés non-triviales des AC sont généralement indécidables, cela réduit immédiatement votre programme et vous avez de petits AC intéressants qui produisent de la complexité élevée. Cela signifie en un sens "Bien, j'accepte de rétrograder mon grandiose intérêt et je me contente d'observer le local au lieu du global."

Stephen Wolfram : Je pense que c'est le cas si on arrive à mettre la main sur une théorie fondamentale de la physique que ...

Greg Chaitin : Tu vas y arriver, Stephen, tu vas y arriver ...

Stephen Wolfram : Bon, merci ... Greg pense que c'est juste une question d'effort (des gens rient). Ou alors si je ne fais pas de version 8 de Mathematica, ou quoi que ce soit. Non, revenons à ce qu'on disait, ainsi la

question était : si on trouve une simple règle, sorte de règle ultime pour la physique, peut-être que c'est un programme discret, et qu'on lance cette chose, et si on croit qu'en trouvant cette chose et en la faisant tourner, ça va nous donner tous les détails un par un de l'Univers grâce au déroulement de ce programme. Peut-être n'est-ce pas pertinent en pratique, mais je ne vois pas comment l'infini intervient là-dedans ...

Cris Calude : Mais si vous le faites tourner pendant un temps très court, vous allez juste avoir une description très primitive, en laissant définitivement de côté de nombreuses propriétés intéressantes de l'Univers. Comment savoir, par exemple, s'il existe une limite finie à la course de ce programme ?

Stephen Wolfram : Mais ce programme n'est pas un modèle, au même sens où l'est le modèle de Navier-Stokes, au sens de l'analyse numérique. Ce programme serait une représentation précise de l'Univers à chaque seconde, et pour tous les détails et caractéristiques dans l'Univers. Ce n'est pas en le faisant tourner qu'on va arriver de plus en plus près de ce qu'est l'Univers (au sens où vous essayez d'obtenir une meilleure approximation de lui). Non ! Si vous le faites tourner pour 10-200, vous allez avoir exactement les premières 10-200 secondes de l'Univers. Si vous le faites tourner 10200 fois plus longtemps, alors vous obtenez la première seconde de l'Univers, la seconde entière, je veux dire absolument exactement, juste comme ...

John Casti : ...dans chacun de ses détails ...

Stephen Wolfram : ... c'est ça, dans chacun de ses petits détails, donc je ne vois pas du tout ce que vient faire l'infini ici ! Par exemple, si vous posez la question : "Peut-il survenir dans l'Univers (pour des êtres intelligents, comme nous) une super-intelligence" alors cette question, si l'Univers est illimité, pourrait en principe être une question indécidable, et alors on retrouve là l'infini. Mais quand il s'agit d'une description basique de l'Univers, je ne vois pas comment l'infini intervient dans cette idée de comment la physique va se présenter.

Greg Chaitin : Oui, Stephen, mais Cris dit seulement que l'infini a été pendant un moment une approximation commode pour le fini, car c'était juste que les mathématiques continues étaient plus développées que les mathématiques combinatoires, avec lesquelles il est un petit peu plus dur de travailler, et donc c'est un accident historique. Il dit juste que c'est une approximation commode. Il ne dit pas que l'Univers est continu.

Cris Calude : Encore pire : Je dis qu'il est aléatoire. Ou qu'il est plus productif pour l'esprit humain de le penser comme aléatoire, de façon à ce qu'on essaye d'en extraire le plus d'information possible. Ça vous rend un

petit peu modeste dans ce que vous essayez de réaliser ... et ensuite peut-être ...

vii L'univers est-il calculable ?

Stephen Wolfram : Je pense à une question-clé dont on voudrait certainement connaître la réponse, et que j'ai mentionnée plus tôt : "L'univers est-il comme π ou comme Chaitin's Ω number ?" C'est clairement une question de grande importance si vous êtes concernés par les fondements de la physique.

Paul Davies : Puis-je affiner cela en donnant une métaphore très claire ? Imaginons – en oubliant le principe holographique – qu'on pixelise le Cosmos entier et qu'il y ait un nombre fini de variables qu'on puisse spécifier pour chaque pixelisation. Maintenant imaginons qu'on prenne les digits de π en binaire et qu'on estampille chacun de ces pixels par des nombres à l'intérieur des pixels pour faire comme un gigantesque Automate Cellulaire, on labellise chacun d'eux avec les digits de π , donc on a un chaos mais ensuite on va au niveau suivant des digits de π et c'est encore le chaos, mais quand on a épuisé les digits – on obtiendra cet Univers, c'est sûr : c'est ça, cet Univers ... ou tout autre Univers !

Stephen Wolfram : Bon, mais ça embrouille certaines choses.

Paul Davies : OK, qu'est-ce que j'embrouille ?

Stephen Wolfram : Bon, OK, voilà ce que vous mélangez. Vous pouvez dire que lorsque vous recherchez une théorie fondamentale de la physique, une fois que vous avez un calculateur universel quelconque et si vous croyez que l'Univers est computable, alors vous avez déjà la réponse.

Paul Davies : C'est vrai. Elle est là, c'est juste qu'on ne sait pas où.

Stephen Wolfram : L'embrouille est là. Ce que j'affirme est quelque chose de beaucoup plus extrême, je dis que si on réussit à trouver une loi fondamentale de la physique, je sais que ça va nous donner accès à une certaine loi de l'évolution, donc je dis : cette condition initiale, cette loi de l'évolution va nous donner précisément tous les détails de l'Univers. Donc cela signifie que quand on extrait les digits de π , le premier digit est ce qui se passe dans les premières 10-200 secondes, le digit suivant est ce qui arrive à la suite ...

John Casti : Stephen, suggérez-vous que l'Univers est computable ?

Stephen Wolfram : Oui.

John Casti : Mais alors, je dois demander, pensez-vous qu'il est computable par tous les modèles possibles de computation, ou vous vous confinez au modèle de Turing de la computation ?

Stephen Wolfram : Je dis qu'il est computable au sens de Turing (au niveau de la computation de Turing).

Cris Calude : Donc l'aléatoire n'est qu'une illusion.

Stephen Wolfram : C'est ce que je dis – pour notre Univers. L'aléatoire est une chose fascinante à étudier, sans aucun doute, mais c'est simplement que nous n'en disposons pas ... nous ne pouvons pas le produire. C'est juste qu'il n'est pas fabricable dans notre Univers.

Greg Chaitin : Mais nous avons du pseudo-aléatoire.

Stephen Wolfram : Oui, nous avons du très bon pseudo-aléatoire.

Greg Chaitin : Comme les décimales de π . Vous les observez – ça paraît aléatoire.

Stephen Wolfram : La question concerne votre effort pour calculer Ω : est-il vraiment possible de le faire dans notre Univers ? Si vous aviez plus d'outils, tous les outils qu'on peut construire dans cet Univers : est-ce que Ω est réellement présent dans notre Univers (présent pour qu'on le trouve et qu'on l'observe) ? Et je pense, mais je ne sais pas, car nous ne le saurons pas jusqu'à ce qu'on ait une théorie fondamentale de la physique, d'où mon affirmation – mon hypothèse de travail, si vous voulez – qu'il n'y a aucun moyen d'extraire Ω de notre Univers, et que le véritable aléatoire, au sens de Greg, n'existe simplement pas dans notre Univers.

Cris Calude : Comment, sous cette hypothèse, réconciliez-vous le fait que, dans cet Univers, on ne puisse pas extraire ce type d'information avec le fait que, dans cet Univers, on puisse parler de choses qui soient si opaques ?

Stephen Wolfram : Je pense que c'est un fait fascinant qu'on puisse parler de l'infini alors qu'on ne peut pas compter jusqu'à l'infini, et qu'on puisse avoir, même dans Mathematica, le symbole pour l'infini, il a des propriétés variées, et on peut raisonner sur l'infini. Je pense que c'est un sujet très intéressant ...

Greg Chaitin : On peut parler des Licornes aussi ! (tout le monde rit.)

Stephen Wolfram : Ce que je veux dire c'est qu'il est commode de déclarer que notre Univers est quelque part dans π , et de dire exactement là où il se situe – pour des raisons/motifs algorithmiques – le fait de dire

exactement où contient autant d'information que l'Univers entier en donne concrètement en tant que programme ... Ce que je dis, c'est que tout l'Univers pourrait tenir dans le creux de la main, comme un petit morceau de code Mathematica, puis il suffirait de le faire tourner et vous fabriquez l'Univers entier. Et d'autre part, ce n'est pas une question triviale, ou bien cela est concevable, ou bien il faut un plus haut niveau de computation au-delà du modèle de computation de Turing, ou quelque chose d'autre ...

Greg Chaitin : Il me semble que votre point de vue est que la computation de Turing est un a priori – que cette notion de computation est donnée avant qu'on commence à discuter de l'Univers physique. La position de David Deutsch est que ce qu'on considère être une machine de Turing universelle ou une computation possible, dépend complètement de la physique de notre Univers et pourrait être totalement différente du modèle de Turing.

Cris Calude : Mais vous avez alors une circularité !

Greg Chaitin : Oui.

Stephen Wolfram : C'est l'une des choses remarquables – je veux dire, le fait que nous ayons découvert au siècle dernier l'invariante idée de ce qu'est la computation ... bien, si on a raison au sujet de cette idée, c'est juste un bel accident historique. Je veux dire que ça pourrait bien être le cas – et ce qui semble être enseigné en Physique le laisse supposer, car tous les 60 ans quelqu'un dit que tout ce qu'on savait en physique jusqu'à maintenant est faux, et qu'il y a un autre niveau de la matière en jeu en physique, et d'après cette intuition on pourrait dire : "De même, dans 60 ans, tout ce qu'on sait sur la computation sera faux et un autre niveau de computation sera possible" On pourrait dire : "Tout ce qu'on savait sur les mathématiques va changer aussi et dans 60 ans un autre type de mathématiques sera possible !" Mais on ne pense pas que ça va arriver, n'est-ce pas ?

Paul Davies : Bon, faux sera toujours faux, non ? J'espère ...

John Casti : Il serait plus précis de dire que tout ce qu'on fait est seulement une approximation de la chose réelle, et le prochain niveau possèdera toutes sortes de propriétés différentes et ainsi de suite. Mais je pense que cet argument que vous venez juste de formuler, si c'est vrai qu'on va continuer à avoir une approximation, puis une meilleure approximation, et une encore meilleure et ainsi de suite – et on arrivera jamais à la fin – alors c'est un argument joliment puissant contre la théorie ultime de la physique, non ?

Stephen Wolfram : Non, voilà ce que je suis en train de dire : les observations empiriques des quelques derniers siècles ont été telles qu'il semble que ce soit un puits sans fond, et nous sommes apparemment incapables de

même espérer ce qu'on fera. Je ne pense pas que ce soit correct. Je ne pense pas que ce soit la correcte intuition. Je pense que la raison pour laquelle les gens ont tendance à avoir cette intuition imprécise est que les gens n'ont pas vu d'exemples où de petites choses simples peuvent construire des choses aussi riches que ce qu'on pourrait imaginer être l'Univers. Regardez, ça vaut le coup de faire le parallèle avec les mathématiques. Si on observe les mathématiques, on peut se demander : les mathématiques sont-elles un même genre de puits sans fond ? En gros, est-ce que les mathématiques d'aujourd'hui vont être nécessairement remplacées par les mathématiques de demain, et est-ce que les problèmes non résolus d'aujourd'hui seront nécessairement résolus demain – dans les mathématiques de demain ? Cela semble être un problème analogue, et mon affirmation se formulerait ainsi (et je pense que Greg serait d'accord avec ça au moins, mais peut-être pas sur les autres points), si vous observez les mathématiques, et l'un de mes exemples favoris c'est les équations diophantiennes, des équations d'entiers : dans l'Antiquité, ils ont appris à fabriquer des équations diophantiennes linéaires. Plus tard au 19ème siècle, ils ont appris à faire des équations diophantiennes quadratiques. Puis au début du 20ème siècle, ils ont appris à faire des équations diophantiennes elliptiques qui sont en train de nous conduire aux équations diophantiennes cubiques, et ainsi de suite. mais il y a encore quantité d'équations diophantiennes qu'on ne sait pas résoudre, et la question est : si on se projette dans le futur des mathématiques, allons-nous, tout comme (visiblement) pour le futur de la physique, être capable d'avancer dans le monde des équations diophantiennes ? Alors ma pensée est : non, ma pensée est qu'on fonce dans le mur, qu'on est vraiment au bord de là où on peut aller, et tout ce qui se trouve au-delà de cette limite, va probablement se révéler indécidable. Et on sera dans la situation où le niveau ultime de la connaissance devient quelque chose comme une nécessité logique, en relation avec les limites ultimes de la connaissance ... plutôt que ... c'est un domaine où on va vraiment toucher le bout... et ... peut-être qu'on ne le voit pas ... il faudrait peut-être voir maintenant ce qu'en pensent les autres.

Greg Chaitin : Vous avez employé le mot "embrouillé" (muddle-headed). C'est un très bon mot, car si vous voulez éliminer le caractère embrouillé, alors NKS s'impose !

Stephen Wolfram : C'est une affirmation intéressante.

Greg Chaitin : Laissez-moi dire pourquoi. Car parlons de l'idée de hasard véritable dans l'Univers physique. Cela signifie qu'il existe des choses qu'on ne peut pas comprendre, des choses irrationnelles. Maintenant ce que je comprends de vous et du principe de raison suffisante de Leibniz (auquel Gödel croyait aussi) est ceci : si vous croyez que tout est rationnel et que

tout dans l'Univers devrait être compréhensible en utilisant la raison, alors je pense que vous êtes nécessairement poussé à vous tourner vers un Univers discret, dans une direction ... hautement déterministe ...

Stephen Wolfram : Oui, ça pourrait être déterministe ... oui, mais il se pourrait aussi que ce soit un autre niveau de raisonnement. Peut-être que le niveau de Turing de raisonnement n'est pas le bon niveau de raison pour notre Univers. Leibniz ne connaissait pas le niveau de Turing du raisonnement, il possédait juste un niveau intuitif de raisonnement, nous pensons qu'il avait un sens précis de raisonnement qui est celui qui ... au jour d'aujourd'hui, il semble être suffisant pour décrire ce qu'on fait en computation. Je pense au fait qu'on puisse décrire des choses qu'on est incapable de fabriquer, c'est un point intéressant, c'est une sorte de clé du succès pour les mathématiques symboliques. mais il y a une question qui est loin d'être claire, au sujet de jusqu'où peut-on aller dans la description des choses qu'on ne peut pas faire ? Voilà, est-ce que c'est le cas – Je sais que dans la théorie des ensembles, les gens se sont toujours inquiétés de savoir si on commence à nommer des types de choses de plus en plus infinies, et encore plus, est-ce que cela ne va pas finir par se casser complètement la gueule ? Peut-on continuer à décrire des choses quand on arrive à des niveaux d'infinis dont on ne peut plus parler ?

Greg Chaitin : Peut-être n'y aura-t-il pas assez d'infinis pour y aller ?

Paul Davies : Qu'en est-il de l'hyper-computation, qui est quelque chose qu'on peut imaginer mais qu'on ne peut jamais faire ?

Stephen Wolfram : Je pense que c'est une question de physique, ou bien l'hyper-computation est quelque chose qu'on peut réellement imaginer sans jamais pouvoir la faire, ou bien l'Univers permet l'hyper-computation ...

Paul Davies : Ah, alors on revient sur cette affaire de ce qui est faisable en principe mais peut-être non réalisable en pratique parce que c'est en dehors des capacités de l'Univers de nous fournir ce qu'il faut par la technologie pour qu'on puisse un jour y arriver.

Stephen Wolfram : C'est ça.

Paul Davies : Donc, il y a des choses qui sont logiquement possibles, mais ce n'est pas réalisable car c'est au-delà des capacités de la physique.

Stephen Wolfram : C'est juste. Ainsi votre question de savoir s'il existe des choses que nous ne pouvons pas réaliser dans notre Univers – ce pourrait être une réponse à la question de ce qui est réalisable. Mais je pense que Karl (Svozil) a préparé quelques notes et veut dire quelque chose.

Karl Svozil : Je voudrais d'abord faire un commentaire sur les problèmes qui ont été évoqués auparavant : fini contre infini, continu contre discret, aléatoire contre déterministe. Ce faisant, je voudrais me servir d'un concept (ou d'une approche) qu'on peut tirer de la psychologie freudienne et qu'ils appellent "attention suspendue", mais avant de commencer, je veux signaler que je suis plutôt du côté des idées de Stephen, et que moi aussi je pense que l'Univers est discret car cette idée est plus iconoclaste. D'un autre côté, je suis aussi très intéressé par la technologie du continuum car il doit y avoir une sorte de correspondance avec les puissances de l'infini qui devraient donc être en ligne avec ce continuum. L'exemple parfait pour moi, et depuis longtemps, serait le paradoxe de Banach-Tarski, qui permet la création et le morphing d'objets en réarrangeant un nombre fini de parties, par l'intermédiaire de cartes préservant les distances (essentiellement des rotations et des translations). Cela pourrait être un exemple de ce que seront un jour les technologies. Donc, l'un dans l'autre, je pense que nous devrions débattre de cela de façon triple : perspective scientifique, perspective métaphysique et perspective historique. Il faudrait couvrir ces trois aspects, et je vais commencer en parlant d'une chose qui m'intéresse en ce moment. Je pense que, pour aller au-delà de ce dont nous discutons pour l'instant, l'observateur doit s'intégrer dans le système. Cela m'excite beaucoup ! J'adorerais vivre pendant une période beaucoup plus longue que ce que je vais pouvoir en vouloir, car j'aimerais voir ce qui va arriver – et je me demande quelles seront les nouvelles idées et théories qui vont venir après.

Stephen Wolfram : Puis-je essayer de poser une question qui attise ma curiosité ? Bon, disons qu'un jour, on dispose de lois fondamentales pour l'Univers. On les connaît, on les a, on peut les écrire sous forme de code, dans Mathematica ou autre. Alors – quelle genre de choses pourrait-on en conclure ? Que se passe-t-il une fois qu'on sait que notre Univers tient en six lignes de code Mathematica et qu'on les a sous les yeux – qu'est-ce qu'on ferait de ça ? Quelles conclusions en tirerons-nous ?

Paul Davies : Puis-je ? Je pense que la première question à se poser est clairement de se demander si c'est le seul, ou si c'est juste l'un parmi tant d'autres pouvant être obtenus, peut-être différents et en nombre infini, en d'autres mots existe-t-il d'autres façons ou est-ce la seule ? et je suppose que la réponse doit être "Non". Ainsi la question devient : est-ce la plus simple façon de le faire en tenant compte de l'existence d'observateurs tels que nous ?

Stephen Wolfram : Que signifie 'plus simple' ? Je veux juste dire qu'on se sentirait très spécial si on était le plus simple Univers pouvant être fabriqué.

Paul Davies : C'est vrai, ce serait plutôt arbitraire ...

viii La chase à l'univers qui est le nôtre

Stephen Wolfram : Ma supposition est que là, dans l'univers des Univers possibles, il y en a des zillions qui sont assez riches pour supporter une certaine sorte d'intelligence comme la nôtre, et qu'on l'écrive en *Mathematica*, en Java ou autre – ce qui aboutirait à des agencements différents, dont l'un d'entre eux serait le plus simple Univers, ou ça reviendrait au même, cela reste, bien sûr, à voir. Ainsi une possibilité serait que ce soit un Univers ayant l'air simple, car nous vivons dans cet Univers-là et les primitives que nous injectons dans nos langages informatiques sont telles que notre Univers finira par être simple.

Paul Davies : Qu'on ne puisse pas découvrir notre Univers s'il est plus compliqué. Dis autrement, nous n'arriverons jamais à compléter le programme dont vous parlez à moins qu'il soit suffisamment simple, bien sûr, pour nous. Donc par définition, ou c'est un mystère, ou c'est simple.

John Casti : Ainsi les autres mondes réels sont trop complexes ?

Paul Davies : Juste.

Stephen Wolfram : Bon ça, OK – donc une possibilité serait qu'il n'y ait aucune autre façon ... non, je ne pense pas que ce soit le cas, je pense qu'on a besoin de décortiquer un peu plus. J'espérais que vous, les gars, allaient essayer d'aider un peu plus ...

Cris Calude : Supposons, supposons par l'absurde que cet Univers est 60 lignes de Mathematica, comment on se rendrait compte que c'est vrai ?

Stephen Wolfram : De la même manière qu'on le ferait pour 6 lignes. Vous le lancez ...

Cris Calude : Oh, oh, OK ... alors comment vous feriez pour 6 lignes ?

Stephen Wolfram : C'est une bonne question. En réalité, c'est dur ! Je suis sur le coup, je fais concrètement la chasse aux Univers, OK ? Je peux vous dire qu'être en première ligne, c'est pas une mince affaire. Vous fabriquez quelque chose qui est un Univers-candidat, et vous pouvez souvent voir assez vite que ça ne marchera pas. Quelquefois, il y a un candidat, une chose qu'on a attrapé dans nos filets, qui n'est pas si sans espoir que ça, quand cela arrive vous le démarrez, vous le laissez tourner, ça déborde de la mémoire, etc. Maintenant notre espoir est qu'avant d'en arriver là, ils puissent montrer

qu'ils possèdent certaines caractéristiques de notre Univers réel, comme un espace dimensionnel fini. Comme je l'ai déjà dit, mon but est de récapituler l'histoire de la physique dans une milliseconde en étant capable de prendre la chose naturelle et concrète qui se trouve être cet Univers repêché et pouvoir en déduire les lois physiques effectives, puis de les comparer avec les autres ... C'est la seule façon, mais vous avez raison, on ne saura jamais si cet Univers qu'on a attrapé est exactement notre Univers. Ça pourrait marcher, vous savez, on a coché 20 caractéristiques sur la liste des caractéristiques, et elles fonctionnent toutes parfaitement. Mais malheureusement, c'est exactement notre Univers, sauf qu'il manque bêtement le lepton ... Mon opinion personnelle est que si l'Univers est simple, alors il y a des chances pour qu'on ne le manque pas à cause d'un petit bit. Il est très improbable que parmi les Univers simples, il y en ait un qui soit exactement le nôtre à un bug près.

Karl Svozil : Stephen ... Je pense que vous avez laissé de côté qu'il puisse juste y avoir qu'un seul Univers qui soit consistant. Et ce pourrait être notre Univers, le seul consistant ; un Univers créé par computation universelle ; avec peut-être une certaine interface au-dessus.

Stephen Wolfram : Que signifie la consistance ?

Karl Svozil : Consistance au sens logique mathématique.

Stephen Wolfram : Je ne pense pas que ça veuille dire quelque chose.

Greg Chaitin : Pas pour les programmes.

Paul Davies : Ni pour la physique théorique, je pense.

Stephen Wolfram : Je ne pense pas que la consistance ait quelque chose à voir avec les programmes. Je veux dire : vous prenez le truc et vous le lancez, ça peut faire quelque chose que vous n'aimez pas, mais il n'y a pas une telle notion de consistance axiomatique. Je voudrais discuter un peu plus sur ce point ...

Greg Chaitin : Bon, une possibilité : on dispose de génomes entiers aujourd'hui. On a le génome complet de centaines de créatures différentes et les gens les comparent, donc un futur possible serait qu'il apparaisse des tas de candidats, et qu'il soit très très difficile de voir s'ils sont justes ou faux (car ça concerne une échelle de 10-200 ou à peu près), donc on pourrait finir par avoir un domaine futur qui soit de "l'universalité comparative" – on a toutes ces lois pour l'univers qui ne peuvent pas être éliminées et ressemblent aux nôtres, et on les étudie, on en discute, on les compare, ce pourrait être un sujet d'étude futur. Les étudiants devraient passer des examens sur leurs dix univers possibles favoris ...

Stephen Wolfram : Au jour d'aujourd'hui, on ne connaît pas les lois fondamentales de la physique. Ce que vous dites, c'est que dans le futur, on pourrait disposer de la collection entière des Univers fondamentaux possibles ou candidats ...

Paul Davies : ... d'assez bons univers. Et il est facile d'imaginer qu'il y en ait deux qui soient pratiquement indiscernables ...

Stephen Wolfram : Mais je ne pense pas que ce soit si facile à imaginer ! Car si les lois sont suffisamment simples, la distance entre chaque candidat univers est petite ...

Paul Davies : C'est vrai ... Mais il faut faire la supposition de lois immuables – car on peut toujours imaginer d'ajuster des choses ...

Stephen Wolfram : Je pense que ça n'a pas de sens.

Paul Davies : Bon, alors je pense que c'est horrible.

Stephen Wolfram : Non, mais ça ne fait pas sens. Voyez ce qui se passe avec quelque chose comme les automates cellulaires, les gens disent toujours "Ne pourrait-on pas obtenir des comportements plus riches si on laissait les règles changer tandis que l'AC évolue (ou pendant que l'automate cellulaire tourne) ? C'est une confusion. Et la raison pour laquelle c'est une confusion, c'est parce qu'on ne pourrait jamais avoir de comportements plus riches en changeant le hardware sous-jacent car on peut toujours simuler n'importe quelle autre forme de hardware à l'intérieur du hardware déjà à disposition, si votre système est universel.

Paul Davies : Mais connaissez-vous ... les jeux de Parrondo ?

Stephen Wolfram : Non.

Paul Davies : Eh bien, vous avez un jeu de hasard qui a une espérance de perte. Et vous avez un autre jeu de hasard qui a aussi une espérance de perte. Vous pouvez les jouer en combinaison avec une espérance de gain même si la combinaison est aléatoire. Cela montre que changer aléatoirement les règles au fur et à mesure possède concrètement un aspect créatif. C'est un peu comme le changement des règles de vos automates cellulaires, même aléatoirement, qui pourraient vous apporter plus, vous apporter peut-être ...

Stephen Wolfram : Pas si sûr si elles changent aléatoirement ! Car dès que vous parlez d'aléatoire, vous parlez de l'injection d'une certaine information venant de l'extérieur du système ...

Paul Davies : Bon, pas obligé de le faire de cette façon, on peut le faire aussi de façon systématique ...

Stephen Wolfram : Mais dès que vous le faites systématiquement, pourquoi ce ne serait pas juste un bit supplémentaire dans les règles ?

Greg Chaitin : Pouvons-nous revenir à votre question d'origine ? Vous disiez donc qu'on aurait les lois de l'Univers en face de nous. Et alors que se passe-t-il ? Un scénario possible est le suivant : disons que cette loi est très simple ; alors on pourrait penser avoir l'Univers le plus simple possible, qu'il y a un certain type de nécessité logique qui le pousserait à seulement être comme ça, etc. Mais supposons que vous trouviez les lois de l'Univers et qu'elles soient vraiment compliquées. Pas compliquées au point de ne jamais pouvoir les trouver, mais qu'elle s'avèrent être vraiment compliquées. Au point de penser que tous les univers possibles puissent exister, sinon pourquoi devrions-nous être dans cet Univers compliqué, plutôt que dans un plus simple qui était une possibilité mais qui n'est pas devenu notre Univers, cela a donné un genre différent d'Univers, mais d'une certaine façon, c'est un bon Univers aussi – même si ce n'est pas le nôtre. Est-ce concevable que ...

Stephen Wolfram : Oui, je pense que s'il s'avère que ça ressemble à notre énumération des Univers, celui-ci serait le tout premier qui serait vraiment semblable – même si c'est une sorte d'illusion, car cela dépend des types de primitives qu'on choisit ...

Greg Chaitin : ... dans le classement par complexité, oui ...

Stephen Wolfram : Mais dans ce cas, ce serait vraiment comme s'il y avait quelque chose de logique et d'inévitable là-dedans ...

Greg Chaitin : Et peut-être que nous sommes uniques, alors, peut-être est-ce le seul Univers.

Stephen Wolfram : Mais si nous sommes juste la planète aléatoire N° 3 d'une étoile au bord d'une galaxie aléatoire, ce serait alors comme s'il devait y en avoir beaucoup plus comme nous, non ? Je ne sais pas, je suis juste curieux – car je n'ai pas encore compris cela, je n'ai pas encore mis à plat ce problème. Disons que si l'Univers est raisonnablement simple dans notre représentation et supposons qu'il y a 10 sortes de candidats-Univers sur la zone où on penserait localiser notre Univers, mais ils possèdent dix-huit dimensions et demie d'espace et supposons qu'ils aient des problèmes variés qui fassent qu'ils ne tombent pas vraiment justes ... donc qu'en concluons-nous, que dirions-nous, on serait alors ...

Greg Chaitin : ... dans ce cas on pourrait croire que tout ça est arbitraire et alors l'idée d'un ensemble ou d'un paysage constitué d'univers alternatifs acquiert un certain poids, commence peut-être à avoir une certaine réalité, à sembler plus convaincant. Alors que si on a un univers très

simple et qu'il n'y a pas d'autres candidats valables aux alentours, alors on va commencer à penser qu'on est unique, qu'on est le seul univers, ou la seule possibilité – en d'autres mots, qu'il existe une nécessité logique ...

Stephen Wolfram : Puis-je tenter une question de plus – qui, je pense, peut susciter l'intérêt de tous ici – qu'en est-il des mathématiques ? J'ai une position tranchée sur cette question, sur le degré d'inévitable de nos mathématiques. J'imagine que tout le monde ici sait que j'ai une position très déterminée là-dessus, en l'occurrence qu'elles n'ont rien du tout d'inévitable, et que c'est juste un accident historique à la base. Mais je suis assez curieux de savoir si d'autres personnes partagent ce point de vue – ou non.

John Casti : Qu'est-ce que vous entendez par être inévitable – l'inévitabilité des mathématiques – dans le cadre axiomatique, ou ... ?

ix Le futur des métamathématiques

Stephen Wolfram : Donc ce que je dis, c'est que les axiomes des mathématiques, qui nous ont donné les quelques 3.106 articles qui ont été écrits sur les mathématiques, sont seulement quelques-uns des systèmes qui constituent juste l'un des systèmes possibles d'axiomes existant à l'intérieur d'un espace beaucoup plus grand de systèmes axiomatiques, et qu'ils n'ont rien de particulièrement spécial qui les distinguent. J'ai fait cela sur l'algèbre booléenne pour laquelle j'ai essayé de déterminer quel numéro de série de système axiomatique possède l'algèbre booléenne, et la réponse a été environ le 50 000ème système axiomatique. Et le problème est que les gens disent néanmoins "Dieu a créé les entiers naturels" par exemple – c'est quelque chose comme ça qui semble plutôt inévitable. Mais le point essentiel est ...

Cris Calude : Vous n'êtes pas d'accord avec ça ?

Stephen Wolfram : Effectivement. Je suis en désaccord avec ça.

Paul Davies : Moi aussi, je pense que je suis en désaccord avec ça. Car on peut imaginer un type d'univers différent où les nombres entiers naturels ne soient pas naturels du tout.

Stephen Wolfram : Oui, c'est ce que je pense. Et Cris n'est pas ...

Cris Calude : Eh bien ...je ne sais pas ... c'est très peu naturel pour moi de penser qu'un entier naturel va faire son apparition d'une manière non-naturelle. Je peux être tout-à-fait d'accord sur le fait que les différents systèmes axiomatiques que nous utilisons sont entièrement accidentels et probablement vont-ils disparaître assez vite. Mais il existe certains blocs de

construction des mathématiques qui semblent avoir beaucoup plus de raisons de persister que les autres. Et les entiers naturels sont dans cette catégorie

Paul Davies : Mais c'est seulement à cause de la structure physique de notre univers. Je veux dire que David Deutsch a relevé cela depuis longtemps, que la physique sous-tend tout le reste. Je pense qu'une machine de Turing est une machine physique. Donc si on imagine un Univers avec un ensemble d'objectifs différents et une substance entièrement différente pour eux – ce qui est computable dans cet univers-là ne l'est pas dans celui-ci !

Cris Calude : Dans cette quête pour trouver de nouveaux Univers, avez-vous un exemple d'univers factice très très simple où vous savez que la mathématique est radicalement différente ?

Paul Davies : La théorie des topos ? ...

Cris Calude : ...on ne peut pas parler de la théorie des topos sans les entiers naturels !

Paul Davies : Donc on a besoin des entiers naturels !

Cris Calude : Oui, on a besoin des entiers naturels ... Je pense vraiment que cette conviction serait beaucoup plus étoffée si on pouvait construire un Univers très simple dans lequel les mathématiques naturelles ne seraient pas reconnaissables à partir de cet Univers (ou à partir des mathématiques de cet Univers.)

Stephen Wolfram : Il y a donc une question : notre Univers nous donne-t-il nos mathématiques ? Je ne pense pas qu'il le fasse.

Paul Davies : Eh bien, je pense qu'il le fait probablement. Nous vivons dans un Univers possédant un certain type de mathématiques naturelles et où il existe certains types de processus qu'on appelle computables. Ce pourrait être un monde complètement différent ...

Stephen Wolfram : Oui, mais les entiers naturels restent invariants pour tous les objets mathématiquement différents que vous avez énumérés.

Paul Davies : Tout ce que j'ai dit, c'est que, autant que je le comprenne, il semblerait qu'on ait besoin des entiers naturels dans tous ceux-ci.

Stephen Wolfram : Imaginons un monde dans lequel la météo possède un esprit, pour ainsi dire, et que nous existions sous forme d'intelligence gazeuse. Et il y a des choses qui se passent là-dedans, mais rien de vraiment computable – vous voyez, imaginons quelque chose comme ...

Greg Chaitin : Pas de doigts ... c'est ça ! Pas de doigts – pas d'entiers naturels ! ...

Stephen Wolfram : Ce serait une explication – donc une affirmation possible serait que dans tout système axiomatique assez riche pour décrire quelque chose d'intéressant, il existe une copie, quelque chose d'analogue aux entiers naturels ...

Greg Chaitin : La géométrie euclidienne n'en a pas. Elle aussi est complète.

Stephen Wolfram : Mais la géométrie euclidienne est décidable.

Greg Chaitin : Oui, elle est complète. Elle est décidable. Pas d'entiers naturels.

Stephen Wolfram : OK, ça va devenir intéressant. On peut faire une affirmation précise disant ceci (un moment de réflexion) : il faut se pencher sur le côté technique. Dans les théories des mathématiques, on peut avoir des mathématiques fermées et décidables, où les déclarations vraies peuvent s'établir de façon finie et ainsi de suite. Ainsi, l'algèbre booléenne en est un exemple, la géométrie euclidienne est un autre exemple de celles-ci, l'algèbre réelle est aussi un exemple, les entiers ou les équations mathématiques entières de Peano (et ainsi de suite) ne sont pas des exemples – eux, ce sont des choses ouvertes qui sont indécidables, comme le sont la théorie des ensembles et d'autres systèmes axiomatiques des mathématiques. Ainsi la question pourrait être (voyons voir, c'est peut-être une véritable évidence) l'affirmation serait qu'à l'intérieur de toute théorie indécidable en mathématiques, vous pouvez créer des objets qui sont semblables à des entiers (ou qui ont les propriétés des entiers). Je pense que c'est une vraie évidence – car je pense que c'est l'universalité qui supporte cet argument, je veux dire que les axiomes de Peano pour l'arithmétique définissent un certain système axiomatique permettant de prouver des choses au sujet des entiers, mais je pense que dès que vous avez un système axiomatique (en fait, ce n'est probablement pas totalement évident), donc si vous avez un système axiomatique qui est universel, dans lequel vous pouvez supporter la computation de Turing – alors j'affirme qu'il s'en suit que vous pouvez construire, vous pouvez simuler à l'intérieur de ce système axiomatique, l'arithmétique de Peano ...

Cris Calude : Vous pouvez simuler les entiers naturels de façons variées, ainsi la différence entre notre Univers et ce que Greg disait sur la géométrie, est qu'on peut simuler les entiers naturels dans les deux Univers, mais avec différentes propriétés. Ainsi, les propriétés en géométrie sont beaucoup plus faibles que celles dans l'arithmétique de Peano – c'est pourquoi on a ce degré de décidable et indécidable.

Stephen Wolfram : Mais je pense que l'affirmation serait que dès que le système est "universel", on sera capable de le simuler ou de le construire avec

les axiomes de Peano . Maintenant, il y a quelque chose qui se passe peut-être dans les aspects techniques de la théorie de la computabilité, mais dans le cas des systèmes pouvant être des calculateurs universels, il y a cette possibilité d'avoir des choses pour lesquelles il existe des propositions indécidables. On affirme – je ne pense pas que ce soit vraiment exact mais – on affirme souvent qu'il pourrait y avoir des choses où on ait de l'indécidabilité mais pas d'universalité. Ainsi c'est le cas du théorème de Friedberg-Muchnik qui est une partie plutôt abstruse de la théorie de la computabilité. Donc la question serait : "Pourrait-il y avoir des systèmes axiomatiques où on ait de l'indécidabilité mais où on ne puisse pas incorporer ... [ne finit pas car Chaitin commence à parler pendant la question] ?"

Greg Chaitin : Je soupçonne que la réponse est : "Oui" mais qu'elle requiert, ou est associée, à une certaine autre construction, comme le théorème de Friedberg-Muchnik. En fait, est-ce que ça ne s'obtient pas immédiatement ?

Stephen Wolfram : Oui, c'est ça.

Greg Chaitin : Le Friedberg-Muchnik nous donne immédiatement un corollaire disant que vous avez un système axiomatique qui est indécidable mais qui n'est pas équivalent à ...

Stephen Wolfram : Je ne pense pas que ce soit juste, car je pense que si on déconstruit – ainsi mon affirmation au sujet du théorème Friedberg-Muchnik est que ... ça devient très technique ! ...

Greg Chaitin : ... pour une grande partie de l'audience, oui ...

Stephen Wolfram : ... donc je dis que si vous regardez à l'intérieur de la construction, on trouve un calculateur universel en train de tourner pas loin ...

Greg Chaitin : Bon, oui, oui – mais vous ne pouvez pas l'obtenir grâce à ce calculateur universel. C'est le point essentiel. C'est là mais vous ne pouvez pas l'obtenir.

Membre de l'audience : Non, le théorème de Friedberg-Muchnik dit que [...] si vous prenez deux ensembles qui soient tous deux intermédiaires – mais si vous prenez les deux ensembles en même temps, ils sont complets ...

Karl Svozil : Ca vaut le coup de mentionner qu'il existe une sorte d'indécidabilité "générée-par-le-calculateur" liée à la complémentarité d'Heisenberg. Cela fait référence à une situation quand on est à l'intérieur d'un système et qu'on décide de vouloir mesurer A mais on ne peut pas mesurer B, ou vice-versa, on veut mesurer B et on ne peut pas mesurer A ... c'est quelque chose d'entièrement impossible "de l'intérieur" ...

Stephen Wolfram : Est-ce vraiment le même phénomène ?

Karl Svozil : Oui, cela conduit à un modèle pour formaliser la complémentarité, et ce faisant Edward Moore a créé la théorie des automates finis, un champ entier de l'informatique, dans les années 1950 ...

Greg Chaitin : 1956.

Stephen Wolfram : Mais la théorie de automates s'en est allée dans une direction totalement différente.

Karl Svozil : Oui, mais cette chose est robuste et me rappelle la remarque de Greg au sujet des univers comparatifs : que ce devrait être un nouveau sujet de recherche.

Greg Chaitin : Oui, c'est l'article de E. F. Moore "Gedanken Experiments on Sequential Machines" publié en 1956. Merveilleux volume intitulé "Automata Studies".

Cris Calude : Shannon était un des éditeurs.

Greg Chaitin : Oui, Shannon et McCarthy étaient éditeurs, von Neumann avait fait une contribution – ce bouquin est fantastique, c'est le début de l'universalité théorique, c'est le premier livre de ce genre. J'en ai vu une recension dans Scientific American – une recension très perspicace – et je me suis précipité dans la librairie, je me suis servi de l'argent de mon repas pour en acheter un exemplaire, et ensuite j'ai commencé à faire des expérimentations de calcul avec le matériau de Moore. Le livre de Moore est épistémologique, la théorie des automates est partie malheureusement dans des orientations algébriques, c'est pour cela que je n'aime pas l'algèbre, mais l'article originel s'occupait de complémentarité et de modèles d'induction [...] épistémologique. C'est un merveilleux papier, et je l'ai rencontré une fois, il avait un autre article – je pense qu'il est probablement mort aujourd'hui, mais je me rappelle bien de lui, il était rouquin, (à lui-même : je pense que c'est une information utile (les gens rient)) – et il avait un autre papier sur l'auto-reproductibilité des Automates Cellulaires...

Stephen Wolfram : Ah, ainsi c'est le gars qui était ...

Greg Chaitin : ... le Gardien de l'Eden

Stephen Wolfram : Oui, mais il y avait aussi des poissons auto-reproducteurs dans l'océan ...

Greg Chaitin : D'accord.

Stephen Wolfram : Il avait fait ce papier dans Scientific American où il disait que, à un moment donné dans le futur, il y aurait de l'auto-reproduction ...

Greg Chaitin : ...des plantes artificielles auto-reproductibles, oui ! Qu'on moissonnerait alors et elles se répandraient partout en mer ...

Karl Svozil : Par exemple, j'aimerais bien faire une expérimentation basée sur ces problèmes de complémentarité computationnelle. ... Ma supposition, ou mon affirmation est que dans un univers généré par ordinateur, on expérimenterait la complémentarité exactement comme on a expérimenté la complémentarité de la mécanique quantique.

Greg Chaitin : Un travail très technique.

Stephen Wolfram : Cela pourrait dégénérer rapidement.

Greg Chaitin : Mais cette discussion porte toujours sur le futur des métamathématiques ?

Stephen Wolfram : Oui, c'est lié à la question de savoir si les mathématiques – si les entiers sont inévitables, et quels aspects de nos mathématiques sont inévitables, quels aspects sont juste une caractéristique de l'Univers où il se trouve que nous sommes ? Quelle est la relation entre (a) l'Univers où nous sommes, (b) les mathématiques que nous fabriquons, (c) ce que nous sommes et (d) l'espace des mathématiques possibles – c'est tout cela.

Greg Chaitin : C'est une question compliquée ... Bien, quand vous pensez que les mathématiques ne sont que des systèmes axiomatiques, alors tout est très arbitraire, car très artificiel, mais si vous croyez aux mathématiques en tant que type de description d'un monde platonique d'idées qui possède une existence indépendante, alors les mathématiques essayent de le décrire et il y a une certaine nécessité en elles.

Stephen Wolfram : Donc votre question est : est-ce que les mathématiques sont sur, sont au sujet de quelque chose ?

Greg Chaitin : Elles sont sur quelque chose – exactement ! Ou c'est juste de la manipulation symbolique ? Je crois qu'il y a une structure à deux niveaux, et de la façon dont vous la regardez – c'est une structure à un seul niveau ...

Stephen Wolfram : Donc si c'est sur quelque chose, peut-être que c'est sur notre Univers.

John Casti : Mais si vous regardez le théorème de Gödel, ça dit que la théorie des nombres porte sur les nombres – qui sont un composant sémantique irréductible ... ce n'est pas juste symbolique.

Stephen Wolfram : Mais attendez une minute ... ce n'est pas le théorème de Gödel ... est-ce que ça ne dit pas que la théorie des nombres telle qu'elle est pratiquée dans les axiomes de Peano ... ça dit effectivement que ce

n'est pas nécessairement sur les nombres ... car le point essentiel, c'est que le théorème de Gödel montre ...

John Casti : ... que les nombres ne font pas partie d'une structure symbolique, que les nombres ont effectivement un contenu sémantique. La théorie des nombres est un fait au sujet de quelque chose. Ce n'est pas ...

Greg Chaitin : ... autrement.

Stephen Wolfram : Mais attendez un peu ! Un des résultats du théorème de Gödel est que – donc le théorème de Gödel dit que, étant donné les axiomes de Peano (le système axiomatique supposé décrire les nombres), qu'en un sens tout système axiomatique dit qu'il y a des choses là-dehors, quelque part, dans le domaine platonique (ou n'importe où ailleurs), qui possèdent les propriétés suivantes [...]. Et ensuite l'objectif d'un système axiomatique est de sculpter ces propriétés tant et si bien que la seule chose que le système axiomatique ne puisse pas décrire ...

Greg Chaitin : Mais c'est impossible.

Stephen Wolfram : Eh bien, c'est le point essentiel ... pour la logique, c'est possible.

Cris Calude : Mais les systèmes axiomatiques sont eux-mêmes un accident de l'histoire. Vous savez, c'est ce qu'on a fait dans la dernière centaine d'années et on essaye de tourner ça en religion, mais il n'y a rien d'intrinsèque qui indiquerait qu'on doit rester bloqué sur ces systèmes.

Greg Chaitin : Bon, parce qu'ils sont autodestructifs. Je veux dire que ce que Gödel a fait, d'une certaine manière, aux systèmes axiomatiques formels – il a montré qu'ils sont inutiles, et c'est le résultat de Gödel. Il a travaillé d'après le programme de Hilbert, et il a explosé le programme de Hilbert. Donc où en est-on après ça, et qu'est-ce qui nous reste ...

Cris Calude : Eh bien l'aléatoire – tout est aléatoire !

Greg Chaitin : ... et Gödel dit "intuition".

Cris Calude : Aléatoire. C'est une possibilité. Encore une fois, ce n'est pas l'unique possibilité, bien sûr, mais c'est une alternative – crédible ...

Stephen Wolfram : On devrait dire pour les gens qui ne savent pas, que c'est concrètement un fait assez intéressant : ce qu'établit vraiment le théorème de Gödel, c'est qu'il existe – en plus des entiers ordinaires qui satisfont aux axiomes de l'arithmétique de Peano – une hiérarchie infinie de types bizarres d'arithmétique non-standard ... où même l'addition n'est pas calculable ...

Greg Chaitin : ... qui ne sont pas les nôtres mais qui satisfont aux mêmes axiomes ...

Stephen Wolfram : ... donc le système axiomatique n'est pas une façon adéquate de discuter de ce dont on parle, pour ainsi dire. Mais votre déclaration sur les systèmes axiomatiques – j'ai tendance à être d'accord avec votre déclaration que les systèmes axiomatiques sont un accident malheureux ...

Greg Chaitin : ... et dont on va se rétablir, j'espère ...

Stephen Wolfram : ... et ce que je pense certainement, et peut-être que je suis ici trop en terrain conquis pour qu'on puisse en juger – c'est que les programmes et les choses qu'on peut spécifier par des règles de calcul sont ce à quoi on devrait s'intéresser, plutôt qu'à ces systèmes axiomatiques utilisés comme moyens de décrire les choses ...

Paul Davies : Dans ce cas, cela soulève un problème qui me tient à cur depuis longtemps, qui est que si on suppose que les lois de l'Univers sont computables, alors on peut imaginer différentes sortes d'Univers, dans lesquels différents types de fonctions mathématiques seraient des fonctions computables mais elles n'obéiraient pas aux règles des machines de Turing classiques. Et alors on aurait une définition différente de la computabilité. On a donc alors une boucle auto-consistante : si nos lois sont effectivement computables, étant donné un Univers qui a le bon type de structure physique pour donner le bon type de computabilité pour faire des lois qui le décrivent comme computable (c'est ce que j'appelle une belle boucle auto-consistante) – est-ce que cette boucle est unique ? Ou alors existe-t-il un autre nombre quelconque de boucles auto-consistantes, où la physique, la matière, les lois et la computabilité sont assez bonnes pour exhiber quelque chose du même genre ?

Greg Chaitin : Bon, la réponse est : probablement. Elles existent probablement, mais Stephen et moi croyons qu'elles n'existent probablement pas. David Deutsch croit probablement qu'elles existent, mais Stephen et moi avons tendance à penser que l'idée courante de la computabilité de Turing est un a priori et qu'il est [indiscernable] de l'univers physique, tandis que David Deutsch dit : "Non !" ...

Stephen Wolfram : Non – la notion courante de computabilité de Turing est suffisante pour notre Univers. Mais votre question est une bonne question et une question précise. Y a-t-il un moyen ... c'est à dire – qu'est-ce qui nous fait croire que la computabilité de Turing est une notion carrément robuste. Toutes ces choses diverses qui viennent avec : les modèles potentiels de computation, qui tous semblent équivalents. Qu'est-ce qui s'est passé dans

le monde de l'hypercomputation, ou supercomputation, c'est qu'il y a toute une variété de dénominations et plein de définitions variées, et contrairement à ce qui est arrivé historiquement avec la computabilité de Turing où toutes les définitions semblaient converger, avec l'hypercomputation, ce n'est pas ce qui semble être en train de se passer, en l'occurrence tout le monde va prétendre que sa propre définition est la bonne, mais il me semble qu'on peut parfaitement et valablement se poser des questions ...

Greg Chaitin : ... vous leur offrez nombre d'alternatives. Ajoutez un oracle à une machine de Turing, pour le problème de l'arrêt. Ça vous donne un nouveau type de machine de Turing. Ce type-là, à son tour, possède son propre problème de l'arrêt. Si on ajoute encore un oracle là-dessus, on obtient une machine de Turing d'un degré encore plus haut qui peut calculer encore plus de trucs et vous avez ainsi une hiérarchie infinie – et en fait c'est beaucoup plus que ... seulement cette structure compliquée.

Stephen Wolfram : Donc : vous pouvez fabriquer un modèle consistant de l'Univers, dans lequel on a machine de Turing plus oracle comme constituant fondamental en route dans le système ?

Paul Davies : Rappelez-vous que c'est une machine ... une machine réelle avec des lois réelles ... faite de matière réelle ...

Greg Chaitin : Bon, on peut certainement avoir une idée consistante de la computation, et un nombre infini de possibilités où on peut calculer des choses qui ne sont pas computables dans notre monde, et continuer comme ça. Donc je suspecte l'existence d'univers physiques parfaitement consistants dans lesquels chacun de ces niveaux d'incomputabilité correspond au niveau de computabilité de l'Univers physique suivant. Ainsi, dans cet Univers suivant, on prendrait comme la chose la plus naturelle d'être capable de calculer la probabilité d'arrêt pour nos machines de Turing, mais pas pour leurs propres machines de Turing.

Cris Calude : Dans ce contexte, une intéressante question est quelle quantité de connaissance obtient-on dans ces modèles de computations de plus en plus élevées. Je peux vous dire par exemple que l'aléatoire algorithmique dont nous avons parlé est de niveau zéro, et les nombres Ω sont en un sens les nombres les plus faiblement aléatoires des aléatoires, mais ils possèdent une connaissance compressée maximale en eux. On peut construire des nombres de plus en plus aléatoires, aucun d'eux n'ayant autant de connaissance compressée que le nombre Ω . Ainsi la question n'est pas seulement la quantité de calculs possibles mais avec quel résultat. Quelle connaissance peut-on déduire de computations plus puissantes ?

Greg Chaitin : ... à notre niveau ? Et la réponse est : pas beaucoup.

Cris Calude : ... au niveau de l'aléatoire ...

Stephen Wolfram : Non, mais il y a un problème : vous vous demandez quels types de computations on peut faire dans un type particulier de système, ou quels types de computations sont faciles à faire avec un type particulier de système. Il est question de savoir si dans ce type de système on peut faire beaucoup de computations qui nous intéressent, ou s'il y a des computations qui nous intéressent et qu'on ne peut pas faire. Donc quand vous dites "peu de connaissance est obtenue" ... vous avez seulement obtenu le nombre Ω ...

Greg Chaitin : Bien, ce qu'on peut percevoir à notre niveau, vous savez ... si vous regardez un nombre Ω de niveau plus élevé, l' Ω de notre niveau nous donne en principe une immense quantité d'informations. Mais à partir d'un Ω de niveau plus élevé (avec son oracle pour le problème de l'arrêt pour chaque machine de Turing), cet Ω (contrairement à notre Ω ici-bas, qui nous donne quantité d'informations ici-bas), cet Ω d'ordre plus élevé ne donne absolument aucune information.

Stephen Wolfram : Donc vous dites qu'à notre niveau, on n'a pas grand chose à tirer d'un Ω .

Greg Chaitin : ... d'un Ω d'ordre plus élevé ...

Cris Calude : Laissez-moi vous donner quelques exemples. Prenons Ω , avec lui vous avez besoin de 3800 bits pour résoudre la conjecture de Goldbach. Avec environ 8000 bits, vous pouvez résoudre l'Hypothèse de Riemann ...

Paul Davies : D'où vous sortez ... ces nombres ?

Cris Calude : Je les ai calculés.

Paul Davies : Mais l'Hypothèse de Riemann n'est pas résolue.

Stephen Wolfram : Il a dit "pour la résoudre – si on avait le nombre Ω ".

Cris Calude : Et si vous avez le nombre Ω .

Paul Davies : Oh – si on avait le nombre Ω !

Greg Chaitin : Cris a écrit un gros programme qui pose si l'Hypothèse de Riemann tient ou pas. C'est comme ça qu'il ...

Cris Calude : Mais si vous disposez de n'importe quelles parties plus élevées d' Ω , vous n'allez pas trouver de nombre fini de bits qui vous donneraient la solution pour l'Hypothèse de Riemann, par exemple – donc en dépit

du fait que vous disposiez d'incroyablement plus de puissance, ce pouvoir ne vous aide nullement à résoudre quoi que ce soit ...

Stephen Wolfram : Mais cela nous aiderait s'il y avait des problèmes formulés au niveau suivant – alors ce serait utile. Mais l'essentiel est – cela est en relation avec ce qu'on considère être intéressant, [et] à ce sujet nous n'avons pas la possibilité de même imaginer [ce qui est intéressant à poser en tant que question] ...

Cris Calude : On n'en sait rien.

John Casti : Ainsi on revient au problème de l'observateur et de la complexité étant en fait relative à l'observateur.

Stephen Wolfram : C'est vrai.

Greg Chaitin (à Wolfram) : L'important pour moi est que je suis définitivement d'accord avec votre déclaration disant que la notion de computation est plus basique que la logique et que les systèmes axiomatiques, et que c'est une meilleure façon de réfléchir sur les choses. J'ai pu formuler ça dans une nuance différente que la vôtre ... mais je pense qu'on est d'accord là-dessus.

John Casti : Alors si tout le monde est d'accord, on peut mettre un terme à ce débat. Merci à tous.

Fin de la transcription.

Annexe B

Groupe de discussion : Qu'est-ce que le Calcul ? Est-ce que la Nature calcule (et comment) ?

Groupe de discussion, NKS Midwest Conference 2008.
University of Indiana, Bloomington, U.S.A.
2 Novembre 2008.

Participants : Cristian S. Calude, Gregory J. Chaitin,
Edward Fredkin, Anthony J. Leggett, Rob de Ruyter,
Tommaso Toffoli et Stephen Wolfram¹.

Moderateurs : George Johnson, Hector Zenil and Gerardo
Ortiz

Organisateurs : Adrian German and Hector Zenil

Transcription : Adrian German²

Traduction : Bernard François

1. Le Prof. Anthony Leggett a demandé à l'éditeur de mentionner qu'il n'avait pas eu le temps de corriger ses interventions en détail les épreuves de la transcription de ses interventions.

2. La transcription et les commentaires sous forme de notes en bas de page sont d'Adrian German (Indiana University, Bloomington) et Hector Zenil avec quelquefois l'assistance d'Anthony Michael Martin (UC Berkeley).

i Computation quantique vs computation classique

Adrian German : Au MIT, il y a environ cinq ans, on posa une question à Stephen Wolfram à la fin de sa présentation³ :

Question (enregistrée) : “Comment fait votre Principe d’Equivalence Computationnelle (PCE) pour expliquer la différence de complexité entre un automate cellulaire (AC) classique et un AC quantique ? ”

Stephen Wolfram (répond à la question dans l’enregistrement) : "Si on prend le formalisme courant de la mécanique quantique et qu’on dise : OK, utilisons-le, supposons qu’on puisse faire des mesures de façon infiniment rapide et que le formalisme standard de la mécanique quantique est une description exacte du monde réel, et pas une sorte d’idéalisaton, comme nous savons qu’il l’est – parce que nous savons qu’en fait, lorsqu’on fait des mesures, on doit prendre un certain petit degré quantique de liberté et l’amplifier presque jusqu’à un nombre infini de degrés de liberté, et d’autres choses du genre – mais supposons qu’on ne s’occupe pas de ça, acceptons l’idéalisaton comme ce formalisme le suggère, qu’on fait une mesure et que ça marche. Alors la question devient : comment relier ce dont je parle avec la computation qu’on peut obtenir dans un cas quantique comparé à un cas classique. Et la première chose que je devrais dire, finalement, c’est que la conséquence de mon PCE est que, dans notre univers, il n’est en fait pas possible de faire des choses qui soient plus sophistiquées que ce qu’un automate cellulaire, par exemple, peut faire. Donc, s’il s’avère que des ACs quantiques (ou des ordinateurs quantiques en quelque sorte) peuvent faire des choses plus sophistiquées que n’importe quel AC classique, alors mon PCE serait tout simplement faux."

Question (enregistrement) : “Par *plus sophistiqué*, vous voulez dire avec une meilleure complexité, ou bien alors dans un sens universel ?”

Wolfram (enregistrement) : "Alors – pour être plus spécifique, une question serait : est-ce que ça peut faire plus qu’une machine de Turing, est-ce que ça peut briser la Thèse de Church ? Une autre serait : est-ce que ça peut faire une computation NP-complète dans un temps polynomial ? Sur ce deuxième point, je ne suis pas sûr. Mais sur le premier point, je vais affirmer très clairement que dans notre monde réel, on ne peut pas faire de computations plus sophistiquées que celles pouvant être obtenues avec une machine

3. Enregistrement tiré de <http://mitworld.mit.edu/video/149/>, dans lequel le paragraphe sélectionné apparaît à la marque 1 :28 :28. La bande-son démarre.

de Turing classique standard. Maintenant, ayant dit cela, il se pose déjà une question technique qui est : si vous acceptez les idéalizations standard de la mécanique quantique, avec des amplitudes complexes possédant un nombre arbitraire de bits en elles, et d'autres choses de ce genre, est-ce que même à l'intérieur de ce formalisme on peut faire des computations qui seraient plus sophistiquées que ce qu'on peut faire dans une machine de Turing universelle standard classique ? Je pense que les gens sentent généralement qu'on ne peut probablement pas le faire. Mais il serait intéressant d'être capable de le démontrer. Par exemple, ce qui suit est vrai : si vous dites que vous faites des computations juste avec des polynômes, il existe alors un résultat datant des années 1930 disant que toute question posée sera décidable, dans ce cas. C'était le résultat trouvé par Tarski. Donc, si les types de primitives que vous avez dans votre théorie n'étaient que des primitives polynomiales, la conséquence serait donc que même en utilisant ces amplitudes de précision arbitraire de la mécanique quantique, vous ne pourriez rien calculer de plus que ce qu'on peut calculer classiquement. Mais dès que vous autorisez les fonctions transcendentes, ce n'est plus le cas – car par exemple même avec les fonctions trigonométriques, vous pouvez facilement coder une équation diophantienne arbitraire dans votre système continu. Donc si vous permettez de telles fonctions, alors vous pouvez immédiatement faire des computations ne pouvant pas être faites par une machine de Turing classique, et s'il y avait moyen de faire en sorte que la mécanique quantique se serve de ces fonctions transcendentes – ce qui ne me semble certainement pas impossible – alors cela impliquerait que la mécanique quantique, dans son idéalisation habituelle, serait capable de faire des computations au-delà de la limite de la Thèse de Church. On peut aussi se poser la question sur le niveau de NP-complétude des choses, et comme je l'ai dit, je suis moins certain sur ce point. Ce que j'en pense personnellement, spéculation si vous voulez, c'est que si on essaye et si on arrive à éclaircir les idéalizations qui sont faites en mécanique quantique, on va trouver qu'en fait on ne réussit pas à faire des choses plus sophistiquées que ce qu'on pourrait faire avec un système du type machine de Turing classique – mais je n'en suis pas sûr !"

Adrian German : Là où finit cette réponse, commence notre conférence. Comme le calcul quantique continue à générer un intérêt significatif tandis que l'intérêt pour NKS grandit régulièrement, nous avons décidé de nous demander : Existe-t-il réellement une tension entre les deux ? Si c'est le cas – quelle information en tirer ? Qu'en est-il si le conflit est seulement superficiel ? C'est alors que nous nous souvenons d'une citation de Richard Feynman extraite des Messenger Lectures de 1964 à Cornell, publiée ensuite

lors de l'édition du livre "La Nature de la Physique"⁴ :

Cela m'a toujours ennuyé que, selon les lois telles que nous les comprenons aujourd'hui, une machine calculante ait besoin d'un nombre infini d'opérations logiques pour trouver ce qui se passe quelque part, quelle que soit la petitesse de la région de l'espace et quelle que soit la petitesse de l'intervalle de temps. Ainsi, j'ai souvent caressé l'hypothèse que la physique va finir par ne plus avoir besoin de déclarations mathématiques, qu'à la fin la machinerie va être mise à jour, et que les lois vont s'avérer toutes simples, comme l'échiquier avec ses complexités toutes apparentes ."

Le germe de cette hypothèse a peut-être bien été planté par un de nos invités d'aujourd'hui : Ed Fredkin⁵ qui dit que, malgré sa collaboration approfondie avec Feynman, il n'avait jamais été sûr que ses théories sur la physique digitale avaient réellement eu un impact sur le légendaire physicien – jusqu'à ce qu'il l'entende dans les lectures citées ci-dessus.

ii L'idée de computation

GJ/HZ/GO : Je voudrais vous remercier tous d'être venus et aussi pour m'avoir invité à ce qui s'est avéré être une conférence vraiment vraiment fascinante. Vendredi soir, juste après être arrivé ici, j'ai soupé avec mon vieil ami Douglas Hofstadter, qui bien sûr habite à Bloomington, et ça a été l'occasion de me rappeler pourquoi (et comment) j'en suis venu à m'intéresser à la computation sur le tard, de façon abstraite. J'étais déjà très intéressé par les ordinateurs quand le livre de Doug "Gödel, Escher, Bach" est sorti – et je me rappelle, étudiant à l'université, avoir consulté un exemplaire dans une

4. En 1981, Feynman fit une présentation à la First Conference on the Physics of Computation, qui se tenait au MIT, où il déclara qu'il lui paraissait en général impossible de simuler l'évolution d'un système quantique sur un ordinateur classique, de façon efficace. Il proposa un modèle de base pour un ordinateur quantique, qui serait capable de telles simulations.

5. L'interaction entre Richard Feynman et Ed Fredkin a commencé en 1962 quand Fredkin et Minsky étaient à Pasadena, et un soir, ne sachant pas quoi faire, ils se sont "invités dans la maison de Feynman", tel que raconté par Tony Hey. Douze ans plus tard, en 1974, Fredkin était de nouveau de passage à Caltech, cette fois en tant que boursier Fairchild, et il passa un an avec Feynman à discuter de mécanique quantique et d'informatique. "Ils ont eu une merveilleuse année de débats créatifs", écrit Hey, "et Fredkin a inventé la Logique Conservatrice et la Porte de Fredkin, qui a conduit au modèle de l'ordinateur à boule de billards de Fredkin." (Voir <http://www.cs.indiana.edu/~dgerman/hey.pdf>)

librairie sur Wisconsin Av., à Washington DC, et avoir été simplement aspiré complètement dans un tourbillon – pour finalement acheter le livre et le lire. Et en le lisant, j’ai pensé “Ma foi, c’est super, car je suis un journaliste scientifique,” et à ce moment-là, je travaillais pour un quotidien à Minneapolis, “je pense qu’on devrait vraiment faire un portrait de ce Hofstadter. “C’était un bon moyen de découvrir ce domaine et j’ai fait le premier déplacement à Bloomington, parmi tant d’autres qui ont suivi, pour rencontrer Doug, et au fil des années nous avons appris à nous connaître assez bien – en fait, il a corrigé mon tout premier livre “The Machinery of the Mind”.

C’est pendant l’écriture de ce livre que je suis tombé sur une citation de l’un de nos invités de ce soir, Tommaso Toffoli, dont je me suis souvenu tandis que j’assistais aux sessions de cette conférence, et en la regardant de plus près l’autre soir, j’ai décidé de m’en servir comme notre point de départ. C’était un article de 1984 dans *Scientific American*, où Brian Hayes⁶ mentionnait les premiers travaux de Stephen Wolfram sur les AC, mentionnant aussi Norman Packard qui était à l’IAS (Institute for Advanced Studies) à l’époque (il devint plus tard le co-fondateur d’une société de prévisions dans ma ville natale de Santa Fe et il fait maintenant des recherches sur la Vie Artificielle), ainsi que des citations⁷ du Dr. Toffoli disant :

... d’une certaine façon, la Nature n’a jamais cessé de calculer la ‘prochaine étape’ de l’univers, et ce pendant des milliards d’années ; tout ce qu’on a à faire – et concrètement, tout ce qu’on fait – c’est de ‘prendre en route’ cette énorme computation en cours.”

C’est en lisant ça et d’autres choses du genre, que j’ai vraiment été accroché et excité à l’idée de cette computation comme explication possible pour les lois de la physique. Nous parlions d’une façon de commencer ici en nous présentant chacun brièvement, bien que la plupart d’entre nous savent qui est qui et pourquoi vous êtes importants, mais juste dire un mot sur le contexte dans lequel vous avez commencé à accrocher et être excité par cette idée.

Greg Chaitin : Étant enfant, ce que je cherchais, c’était la plus excitante nouvelle idée que je puisse trouver. Et il y avait des choses comme la Mécanique Quantique et la Relativité Générale que je regardais en chemin, et le théorème d’incomplétude de Gödel, mais à la base ce qui fut clairement la grande révolution, c’était l’idée de computation que portait en elle

6. <http://bit-player.org/bph-publications/AmSci-2006-03-Hayes-reverse.pdf>

7. <http://www.americanscientist.org/issues/issue.aspx?id=3479&y=2006&no=2&content=true&page=4&css=print>

la technologie informatique. L'informatique était excitante mais j'étais encore plus intéressé par la profondeur mathématique et philosophique de l'idée de computation. Pour moi, il était alors déjà clair que c'était une nouvelle idée mathématique réellement majeure – la notion de computation est une baguette magique qui transforme tout ce que vous touchez avec, et vous donne une façon différente de penser au sujet de tout. C'est un changement de paradigme majeur au niveau technologique, au niveau des mathématiques appliquées, des mathématiques pures, aussi bien qu'au niveau philosophique fondamental, sur des questions réellement fondamentales en philosophie. Et la partie concernant l'univers physique – cette partie n'était pas du tout évidente pour moi. Mais si vous voulez découvrir le monde grâce à la raison pure, qui se soucie de la façon dont il est concrètement construit ? Donc je dis : concevons un monde de pure pensée, qui soit computationnel, avec les computations comme ses unités de constructions de base. C'est comme jouer à être Dieu, mais ce serait un monde que nous pourrions comprendre, non ? Si nous l'inventons, nous pouvons le comprendre – alors que si on essaye de saisir comment le monde fonctionne, ça va tourner encore une fois à la métaphysique . Donc, apparemment, j'ai pris en route une vague, une des plus excitantes qui soit, la plus grosse que je pouvais voir venir !

Ed(ward) Fredkin : Quand j'étais dans l'Air Force et que j'étais stationné au Lincoln Labs, j'ai eu la chance de rencontrer Marvin Minsky presque tout de suite. Peu de temps après j'ai fait la connaissance de John McCarthy, et tous deux m'ont donné quantité de conseils. Je ne sais pas exactement quand m'est venue pour la première fois l'idée de l'Univers en tant que simulation sur un computer, mais c'était à peu près à cette époque. Je me rappelle quand j'en ai parlé à John McCarthy (ce devait être vers 1960), il m'a dit quelque chose qu'il m'a peut-être répété une centaine de fois au cours de nos longues discussions : "Oui, j'ai eu la même idée", sur l'univers en tant que réalisant des computations. Et je disais "Bon, qu'est-ce que tu penses de ça ?", et il disait "Oui bon, on peut chercher voir s'il y a des erreurs d'arrondis ou de troncature en physique ..." et quand il a dit ça, j'ai immédiatement pensé "Oh ! Il pense que je pense à un IBM709 ou 7090 (je suppose qu'il n'était pas encore sorti) là-haut dans les cieux ..." et me venait à l'esprit une espèce de processus computationnel ... je ne pensais pas à l'erreur d'arrondi ... ou à l'erreur de troncature. Quand j'ai parlé de cette idée à Marvin, il m'a suggéré d'aller voir du côté des automates cellulaires, et je n'avais pas entendu parler d'automates cellulaires entretemps, si bien que j'en étais réduit à trouver un article de ... ou me renseigner comme je pouvais, je me rappelle ne pas être arrivé à trouver l'article que von Neumann avait fait à partir duquel – c'étais vers les années 60 (disons 59 ou 60) – j'étais resté

intéressé par ça⁸. Et dans l'une de mes premières expérimentations sur ordinateur, je décidai de trouver la plus simple règle qui soit symétrique de toutes les façons possibles, ainsi j'en arrivai aux automates cellulaires binaires de von Neumann, à cellules voisines, et pensais "Quelle fonction symétrique est possible?". Il s'est avéré que c'était XOR. Je l'ai donc programmée, tout en étant un peu surpris par le fait que cette simple règle puisse avoir au moins un minimum d'intérêt.

Rob de Ruyter⁹ : Je dois aussi remonter jusqu'à ma jeunesse, ou au moins jusqu'au Lycée, quand il y avait encore tout un tas de mystères autour de la biologie (du temps où j'étais au Lycée, au moins) et on faisait de nombreuses et superbes découvertes, on continue toujours à en faire, sur le fonctionnement de la vie à l'échelle cellulaire. C'était une sorte de monde incroyablement fascinant qui s'ouvrait à moi quand je me suis intéressé à la biologie et que j'ai appris à connaître des mécanismes de plus en plus intriqués. En même temps, bien sûr, on faisait de la Physique – concrètement, en Hollande, on vous les enseigne en même temps au Lycée, contrairement à ce qui se passe ici. Et en physique, vous rentrez en contact avec toutes ces belles lois, descriptions mathématiques de la Nature, et elles vous donnent la réelle impression que vous pouvez la saisir, cette Nature, de façon purement intellectuelle. En ce temps-là, il était déjà évident qu'en partant de cette vision physique où vous pouvez tout décrire très précisément, pour aller jusqu'au mécanisme d'une cellule biologique, il fallait couvrir un énorme éventail de choses pour comprendre comment ça marchait, et nous n'en sommes toujours pas capables pour ce qui est des principes physiques sous-jacents. En même temps, à cet âge-là, quand vous êtes au Lycée, vous passez par toutes sortes de transformations hormonales etc., et l'une des choses que vous commencez à réaliser, c'est que (a) lorsque vous regardez le monde, vous tirez des informations de l'extérieur, mais (b) vous avez aussi votre propre esprit qui peut aussi réfléchir et faire de l'introspection. Et vous savez que cette introspection, vos pensées propres, doivent trouver refuge quelque part, d'une manière ou d'une autre, elles doivent se construire à partir de matière – que ça doit se passer quelque part, d'une certaine façon. Alors vient la question qui m'a toujours terriblement fasciné, encore aujourd'hui, (pour faire court) la question de comment la matière conduit à la pensée, la question esprit-matière

8. <http://www.digitalphilosophy.org/>

9. Rob de Ruyter van Steveninck est Professeur au Département de Physique et Programmes à la Neural Science Indiana University Bloomington. Il est membre de l'American Physical Society. Il a un doctorat de 3ème cycle (1986) de l'université de Groningen, Pays-Bas, et a été en post-doctorat à Cambridge au Royaume-Uni et à l'université de Groningen. Sa spécialité est la compréhension des principes de base supportant le codage, la computation et l'inférence dans le système nerveux sensitif.

et tout ce qui s'y rattache.

J'ai eu assez de chance pour trouver une place dans un département de physique, où il y avait un groupe biophysique, et j'ai pu étudier la question de l'esprit – peut-être un simple esprit, l'esprit d'un insecte, ou au moins des petits morceaux de l'esprit d'un insecte – et j'ai essayé de les étudier de façon hautement quantitative, ce qui m'a fait revenir à la belle description mathématique que vous avez en physique. Ainsi je pense à cette tension entre la matière, comment vous la décrivez dans les formalismes mathématiques, et la pensée – comment on sait introspectivement ce qu'est la pensée, et vraisemblablement, jusqu'à un certain point, que tous les animaux ont une pensée – pensée qui, jusqu'à un certain point encore, prend la forme de computation, bien sûr, c'est le genre de choses qui me pousse dans mon travail¹⁰. Et en tant qu'expérimentaliste, j'ai grand plaisir à voir que vous pouvez faire des expérimentations où il est au moins possible de détacher de toutes petites lamelles de ces questions et peut-être rendre les choses un petit peu plus claires au sujet de l'esprit.

(An)T(h)ony Leggett¹¹ : Je pense que je vais être le vilain petit canard du groupe, car je ne suis pas vraiment convaincu qu'une approche computationnelle va résoudre les problèmes fondamentaux de physique auxquels je m'intéresse le plus. Je pense certainement que ce genre d'idées, comme celles dont on a débattu au cours de cette conférence, sont extrêmement fascinantes et il se pourrait qu'elles soient justes – je ne sais pas. Et le domaine dans lequel je pense qu'on pourrait assez facilement démontrer que cette confluence entre science computationnelle et physique a donné des fruits, est bien sûr le domaine de l'information quantique, et du calcul quantique en particulier. Ce qui se passe dans ce domaine, comme je le disais plus tôt, c'est qu'une approche venant des sciences informatiques vous donne une façon complètement différente de regarder les problèmes, en les posant comme des questions auxquelles on n'aurait pas pensé autrement. Il est intéressant de constater qu'au moins pour la décade 1995-2005, quand l'information quantique a clairement trouvé sa voie, un tas d'articles parus à ce moment-là dans les Physical Re-

10. Rob de Ruyter van Steveninck est le co-auteur de *Spikes : Exploring the Neural Code* (MIT Press, 1999) avec le Prof. William Bialek de Princeton University et d'autres.

11. Sir Anthony James Leggett, KBE, FRS (né le 26 Mars 1938 à Camberwell, Londres, Royaume-Uni), occupe la chaire John D. et Catherine T. MacArthur et est Professeur de Physique du Center for Advanced Studies à l'Université de l'Illinois à Urbana-Champaign depuis 1983. Le Professeur Leggett est largement reconnu comme un leader mondial sur la théorie de la physique à basse température, et ses travaux de pionnier sur la superfluidité ont été récompensés par le prix Nobel 2003 de Physique. Il a donné des orientations de recherche en physique quantique des systèmes dissipatifs macroscopiques, et utilise les systèmes condensés pour tester les fondements de la mécanique quantique.

view Letters auraient pu facilement paraître dès 1964, mais ça n'a pas été le cas. Pourquoi ? Parce que les gens d'alors n'avaient pas la connaissance (ou l'intuition) pour se poser ce(s) type(s) particulier(s) de questions. Donc certainement, je pense que ce fut une interaction très très utile et féconde¹².

Mais je pense que si quelqu'un m'avait dit, par exemple, que tel ou tel problème pouvant aisément être posé en physique, ne pouvait trouver de réponse par ordinateur qu'en mobilisant un nombre de bits supérieur ou égal au nombre total de particules dans l'univers, je pense que je n'aurais pas été impressionné outre mesure. Et ma réaction aurait probablement été "Très bien, et alors ?" Je ne pense pas que cela serve à quelque chose, je ne pense pas que la nature vaille à ses occupations en calculant ce qu'elle a à faire – Je pense tout le contraire ! Mais je crois, peut-être un peu plus fondamentalement, qu'une des raisons pour laquelle je suis un peu sceptique quant à cette entreprise, c'est que je pense vraiment que les questions fondamentales en physique concernent beaucoup plus l'interface entre la description du monde physique et notre propre conscience. Un des cas les plus évidents est l'infâme problème de la mesure quantique¹³, que je considère assurément comme un

12. En 1987, Oxford University Press a publié "The Problems of Physics" d'Anthony J. Leggett, où il écrivait : "Bien que [le principe dit anthropique, la 'flèche du temps', et le paradoxe de la mesure quantique] soient historiquement associés à des champs assez différents de la physique – cosmologie, mécanique statistique et mécanique quantique, respectivement – ils ont beaucoup en commun. Dans chaque cas, il est probablement honnête de dire que la majorité des physiciens sentent qu'il n'y a simplement aucun problème à débattre, alors qu'une minorité insiste non seulement sur l'existence d'un problème, mais aussi sur son urgence. Dans aucun des cas, le problème ne peut se résoudre par l'expérimentation, du moins à l'intérieur du cadre conceptuel dominant – une caractéristique qui en pousse beaucoup à le déclarer comme 'simplement philosophique'. Et dans chaque cas, quand on sonde plus profondément, on finit par se heurter à une question plus générale, en l'occurrence : En dernière analyse, une description satisfaisante du monde physique peut-elle échouer à explicitement tenir compte du fait qu'elle est elle-même formulée par et pour des êtres humains ?"

13. Dans le formalisme de la mécanique quantique selon son interprétation conventionnelle, il semblerait qu'un système ne possède pas de propriétés définies jusqu'à ce qu'on le force à les déclarer en pratiquant une mesure appropriée. Mais est-ce la seule interprétation possible des données expérimentales ? On peut résumer la situation en disant que dans le formalisme quantique, les choses ne se 'produisent' pas, contrairement à ce qui se passe dans la vie de tous les jours. C'est le paradoxe de la mesure quantique. Comme indiqué ci-dessus, certains physiciens voient ça comme un non-problème, alors que pour d'autres, cela ébranle tout l'édifice conceptuel du sujet en question. Différentes résolutions du paradoxe ont été proposées. Il y en a une appelée solution exotique, qu'il faut mentionner par les noms donnés à son interprétation, connue comme l'interprétation d'Everett-Wheeler, ou de l'état relatif, ou des multi-univers. Cette prétendue solution est fondée sur une série de théorèmes formels appartenant à la théorie quantique de la mesure, qui garantissent que la probabilité de deux mesures différentes de la même propriété donnant des résultats

problème très sérieux ; et deuxièmement la question de la flèche du temps¹⁴ : qu'on puisse se souvenir du passé, que le passé influence le futur mais non pas le contraire. C'est quelque chose qui sous-tend certains des aspects les plus fondamentaux de notre existence en tant qu'êtres humains, et je pense qu'on n'y comprend rien, du moins pas en termes physiques, et j'ai personnellement la plus grande difficulté à imaginer comment une approche computationnelle

différents (bien sûr, selon des conditions spécifiées de façon appropriée) est de zéro. L'interprétation multi-univers, comme elle est présentée par ses défenseurs les plus enthousiastes, prétend alors, pour le dire crûment, que notre impression d'obtenir un résultat particulier pour chaque expérience est une illusion ; que des résultats alternatifs possibles continuent à exister en tant que 'mondes parallèles', mais qu'on est assuré, grâce aux théorèmes formels cités plus haut, ne jamais être conscient que d'un monde à la fois, et qu'il est le même pour tous les observateurs.

14. Il est presque trop évident pour le dire, que le monde autour de nous semble montrer une asymétrie marquée par rapport à la 'direction' du temps ; de très nombreuses séquences d'événements peuvent apparaître dans un ordre, mais pas dans l'ordre inverse. Pourquoi cette observation est-elle problématique ? A vrai dire, c'est un problème plus gros qu'il en a l'air. Effectivement, dans les discussions générales sur le sujet, la convention est de distinguer au moins cinq différentes flèches du temps. L'une est la flèche 'thermodynamique', une deuxième est la flèche du temps 'humain' ou psychologique, déterminée par le fait que nous nous rappelons le passé mais pas l'avenir. Une troisième flèche est la 'cosmologique' déterminée par le fait que l'Univers est actuellement en expansion plutôt qu'en contraction. Une quatrième flèche est dite électromagnétique, et cela demande une petite explication. Finalement, pour être complet, je dois mentionner la cinquième flèche, qui est associée avec une classe d'événements rares en physique des particules – les désintégrations de certains mésons violant la symétrie dite CP – et liée probablement à d'autres très petits effets au niveau des particules. Cette flèche est assez différente des autres en ce que les équations pertinentes sont elles-mêmes explicitement asymétriques par rapport au sens du temps, plutôt qu'asymétriques par l'imposition d'un de nos choix de solution. Pour l'instant, les origines de cette asymétrie sont très mal comprises, et le consensus général est que ça n'a probablement pas grand-chose à voir avec les autres flèches ; mais il est trop tôt pour en être complètement sûr. A l'égard des quatre premières flèches, un point de vue qui semble au moins libéré des inconsistances internes évidentes, est qu'on peut considérer la flèche cosmologique comme fondamentale ; que la flèche électromagnétique est alors déterminée par elle, de sorte que ce n'est pas par accident, par exemple, que des étoiles irradient de l'énergie lumineuse vers l'infini plutôt que de l'aspirer en elles ; que, parce que les radiations sont essentielles à la vie, cela alors suffit à déterminer la direction de la différenciation biologique dans le temps, d'où enfin notre sens psychologique du passé et du futur ; et finalement, que la flèche thermodynamique est connectée, dans la nature inanimée, avec la flèche psychologique de la façon décrite ci-dessus. Toutefois, tandis qu'il est facile de penser, à chaque étape, aux raisons pour lesquelles la connexion requise doit tenir, il est probablement honnête de dire que dans aucun cas il n'y a de connexion établie avec un tant soit peu de rigueur, et courageux serait le physicien qui parierait sa vie sur l'assertion, par exemple, que la vie consciente doit être impossible dans un univers en contraction. Ce fascinant complexe de problèmes va sûrement continuer à défier les physiciens et les philosophes (ainsi que les biologistes et les psychologues) pendant encore longtemps." *The Problems of Physics*, 1987, Oxford University Press, pp. 148-157.

pourrait améliorer notre approche sur aucun de ces problèmes.

Maintenant, pourquoi j’assiste à des conférences comme celle-ci, c’est que j’espère pouvoir être convaincu un jour. Donc il va falloir attendre et voir ce qui va se passer.

Cris(tian) Calude : Je pense que je suis probablement la plus conventionnelle et la plus ordinaire personne¹⁵ ici présente. Je m’intéresse à la calculabilité, la complexité et l’aléatoire, et j’essaye de comprendre pourquoi on peut faire des mathématiques et ce qui nous rend capables de comprendre les idées mathématiques. Je suis aussi un petit peu sceptique quant à la puissance du calcul quantique. J’ai été impliqué ces dernières années dans un petit projet pour déquantiser différents algorithmes quantiques, c-à-d construire des versions classiques d’algorithmes quantiques qui soient aussi rapides que leurs contreparties quantiques. Et j’en suis arrivé à croire que le calcul quantique et l’information quantique nous en disent beaucoup sur la physique, plus que de pouvoir devenir des outils utiles à la computation. Finalement, je suis très intéressé à comprendre la nature et la puissance de l’aléatoire quantique, et si un genre d’hybridation entre un PC et une source d’aléatoire quantique pourrait être plus performante. Pouvons-nous dépasser la barrière de Turing, même en principe, avec ce genre de computation hybride, ou non ?

*Tom(maso) Toffoli*¹⁶ : Bon, je suis né d’un accident historique dans un bureau de poste, et ça a peut-être quelque chose à voir avec tout ça parce que c’était exactement organisé comme Ethernet : pressez un bouton et vous jetez à terre une grande quantité de lignes (40 miles) auparavant de haute valeur, donc vous devez écoutez ce que vous tapez et si ce qu’on obtient est

15. Cristian S. Calude occupe la chaire de Professeur à l’Université d’Auckland, Nouvelle-Zélande. Directeur fondateur du Centre pour les Mathématiques Discrètes et l’Informatique Théorique. Membre de l’Académie Europaea. Des recherches en théorie algorithmique de l’information et sur le calcul quantique.

16. Les centres d’intérêts et compétences de Tommaso Toffoli comprennent entre autres : Mécanique de l’Information, Fondements et aspects physiques du calcul, Théorie des automates cellulaires, complexité de l’Interconnexion et synchronisation, Modèles Formels de computation compatibles avec la physique microscopique (uniformité, localité, réversibilité, inertie et autres principes de conservation, aspects variationnels, relativistes et quantiques de la computation). Il a prouvé l’universalité de computation des automates cellulaires réversibles (1977) ; il a formulé la conjecture (prouvée plus tard par Kari) que tous les automates cellulaires réversibles sont structurellement réversibles (1990). Il a introduit la porte de Toffoli (1981), qui fut ensuite adoptée par Feynman et d’autres comme la primitive logique fondamentale de la computation quantique. Il a proposé, avec Fredkin, le premier projet concret de computation à conservation de charge (1980) – une idée qui a été reprise par l’industrie de la basse-puissance ces dernières années. Il a prouvé que les algorithmes dissipatifs pour les automates cellulaires pouvaient être remplacés par des algorithmes non-dissipatifs de treillis de gaz (2006).

différent de ce qu'on tape, ou de ce qu'on s'attend à entendre, alors cela veut dire que quelqu'un d'autre est aussi en ligne. Donc les deux s'arrêtent. Puis les deux repartent à nouveau, au hasard – exactement dans l'esprit d'Ethernet. A part ça, quand j'avais cinq ans, c'était juste après la guerre, ma mère avait déjà quatre enfants et était très occupée à essayer de nous trouver de quoi manger au marché noir et ainsi de suite – donc j'étais laissé seul presque toute la journée dans un endroit situé au cinquième ou sixième étage. Là où on vivait, on avait de longs balcons avec de très longs étendoirs, pour la lessive etc ..., et le fil n'était pas fixé au milieu, les points d'ancrages sur le mur étaient seulement aux extrémités des deux côtés. J'étais comme un singe en cage sur ce balcon, il fallait que je pense, que je bouge, que je trouve moyen de passer le temps d'une façon ou d'une autre. J'avais remarqué une chose quand je tirais et secouais cette corde, j'avais découvert qu'en se mettant plus ou moins à un tiers de la longueur et en secouant d'une certaine manière, le truc finissait par osciller de plus en plus. J'avais découvert ce que vous pourriez appeler la résonance, le blocage de phase et d'autres choses du genre, et j'étais très fier de toute cette puissance libérée par des mains [si minuscules]. Mais j'ai eu aussi une grande frayeur car – comme résultat de ma précoce activité scientifique – le plâtre s'était détaché du mur là où la corde était fixée, et il y en était même tombé sur le trottoir, visible en bas même du sixième étage où j'étais, m'assurant d'avoir des ennuis quand ma mère reviendrait le soir à la maison.

En tout cas, le point que j'aimerais souligner est que "Je ne peux pas comprendre pourquoi les gens ne peuvent pas comprendre" (comme l'écrivait Darwin à l'un de ses amis) " qu'une expérience neutre n'existe pas – il n'y a que des expériences 'pour' ou 'contre' un sujet." En d'autres mots, il faut d'abord poser une question, sinon l'expérience n'est pas très utile. Maintenant, on pourrait dire : "Parfait, mais comment faire pour répondre à la question qui a motivé la mise en place de l'expérience?" Et ce qu'on fait avec la computation, les automates cellulaires et tout le tralala, c'est qu'on est juste en train de fabriquer notre propre univers. Mais cet univers qu'on fabrique, on le connaît, on fabrique ses règles donc les réponses qu'on obtient sont précises. On sait exactement si on peut fabriquer de la vie à l'intérieur de tel AC (comme von Neumann) et on sait si on peut calculer certaines choses, donc c'est un monde fabriqué de façon à avoir des réponses précises – pour ce monde limité et inventé – et fabriqué de façon à nous donner, espérons-le, un certain éclairage sur les réponses correspondant aux questions touchant le monde réel. Ainsi, il faut faire de continuel aller-retours, d'une part entre les mondes que nous connaissons car nous les contrôlons entièrement et pouvons réellement répondre aux questions, et d'autre part les autres mondes

sur lesquels on aimerait bien répondre aux questions. Et j'ai l'impression que la computation, les AC et toutes ces choses sur lesquelles Wolfram a attiré notre attention, seraient juste des tentatives pour tirer le meilleur de ces deux types de mondes. Le seul monde pour lequel nous pouvons répondre aux questions qu'il pose est celui qu'on a fabriqué, et pourtant on désire répondre aux questions du monde dans lequel nous sommes – qui possède toutes ces imperfections, et où la résonance ne va pas donner un pic infini mais va quand même faire tomber le plâtre sur le trottoir, et ainsi de suite.

Et je pense qu'on devrait être capable de vivre avec un pied dans un monde, et un pied dans l'autre.

Stephen Wolfram : Voyons voir, je crois que la question était comment on a été impliqué dans la computation, comment on s'y est intéressé, et intéressé ensuite à tout ce genre de choses. Bon, l'histoire en ce qui me concerne est déjà assez longue : quand j'étais enfant, je m'intéressais à la physique. Je pense que je m'y suis intéressé surtout parce que je voulais savoir comment fonctionne la matière, et à cette époque, c'était dans les années 1960 environ, la physique était la position la plus convaincante d'où chercher des explications sur le fonctionnement de la matière. Quand j'ai eu dix-onze ans, j'ai commencé à lire des livres de physique du collège et ainsi de suite. Je me souviens d'un moment bien précis, j'avais douze ans, un livre de physique en particulier a eu une grande influence sur moi : c'était un ouvrage sur la mécanique statistique qui avait en couverture un genre de bande film prétendant illustrer ce qui se passerait en partant d'un gaz composé de sphères solides ou quelque chose comme ça, qui était très bien organisé au départ, pour arriver à un gaz de sphères solides semblant complètement aléatoire. C'était une tentative pour illustrer le second principe de la thermodynamique. Je pensais que c'était une illustration très intéressante, et lisais ainsi toute l'explication mathématique du livre sur le pourquoi ça devait se passer de cette façon.

Je trouvais l'explication mathématique assez peu convaincante, en tout cas complètement plombée car appartenant à ces explications qui – à la base, on dit la même chose sur le temps – finissent par une déclaration du genre "bon, toute l'explication pourrait très bien fonctionner en sens inverse, sauf que, d'une certaine façon, ce n'est pas comme ça que ça marche." (Rire général) Pas très convaincant, donc j'ai pris une décision : très bien, je vais trouver par moi-même ce qui se passe réellement là-dedans, je ferais volontiers ma propre version de ce petit film. A ce moment-là, j'avais accès à un ordinateur simple, un ordinateur anglais : Elliott 903, des ordinateurs utilisés surtout pour diriger des tanks et autres. J'avais huit mille mots de dix-huit bits de mémoire. Autrement c'était un bon ordinateur, et je commençais à essayer de programmer ce truc pour faire marcher le second principe de la

thermodynamique. A certains égards, l'ordinateur était assez primitif, notamment sur les opérations à virgule flottante, donc j'essayais de simplifier le modèle que j'avais pour ce gaz de sphères solides rebondissantes. Bien des années plus tard, je me suis aperçu que le modèle que j'avais fini par mettre au point, était un système d'automate cellulaire à deux dimensions. Cela n'a pas marché : j'ai simulé ce système d'AC en 2D et (a) je n'ai trouvé absolument rien d'intéressant et (b) j'ai échoué à reproduire le second principe de la thermodynamique. J'avais alors douze ans. A ce moment-là, je me suis dit : il y a peut-être quelque chose d'autre qui se passe, et peut-être que je ne le comprends pas, donc je vais travailler dans un domaine où je pourrai en apprendre sur la matière, où je pense que je pourrai mieux comprendre ce qui se passe. Et le domaine dans lequel je me suis impliqué fut la physique des particules. Par la suite, j'en arrivai à connaître toutes sortes de choses sur la physique des particules, la théorie quantique du champ, et j'ai compris beaucoup de choses, ce qui a fait de moi un opérateur respectable dans le milieu de la physique des particules, à l'époque.

A pratiquer la physique des particules, il arrivait qu'on ait à faire des calculs algébriques compliqués, une chose pour laquelle je n'ai jamais été particulièrement bon. J'étais aussi d'avis qu'on devait toujours essayer de se doter des meilleurs outils possibles qu'on pouvait fabriquer. Et même si je n'étais pas très bon dans des domaines comme l'algèbre, j'ai appris à construire des outils informatiques pour faire de l'algèbre. Et finalement, je décidai que le meilleur moyen de réaliser les types de computations dont j'avais besoin, était de construire mon propre outil pour faire ces computations. Donc j'ai bâti un système appelé SMP, qui fut le précurseur de Mathematica – c'était aux alentours de 1981. Pour construire ce système informatique, je devais comprendre, à un niveau fondamental, (a) comment organiser un large éventail de computations, et (b) comment monter un langage qui puisse décrire ce large éventail de calculs. Et cela a nécessité l'invention de ces primitives qui pouvaient servir à faire toutes sortes de computations pratiques. Tout le projet fonctionnait assez bien, et j'en conclus bientôt que j'étais capable de mettre au point les primitives pour monter un système comme SMP. Puis, pour des raisons variées, quand j'ai commencé à réfléchir de nouveau à de la science de base, ma question a été : "Ne puis-je pas prendre les phénomènes qu'on observe dans la nature, et les associer à des types de primitives inventées qui décriraient comment ils fonctionnent – tout comme j'ai réussi à inventer les primitives pour ce langage informatique que j'ai construit ?" Je pensais vraiment que ce serait un moyen d'arriver à faire des modèles physiques tout-à-fait traditionnels, et ce n'est qu'accidentellement que plus tard j'en arrivai à pratiquer cette science naturelle consistant à explorer abstraitement ce que

les programmes informatiques montés par mes soins, faisaient. Et le résultat de ceci fut la découverte de toutes sortes de phénomènes intéressants dans les ACs et ainsi de suite.

Cela me poussa à considérer plus sérieusement l'idée d'utiliser la computation et les programmes informatiques comme moyen de modéliser beaucoup de choses. Je pense par exemple, en termes de conviction, que l'univers peut être représenté par de la computation, chose que je n'ai pas encore prouvée, c'est sûr. Je suis seulement devenu de plus en plus convaincu que c'était plausible – parce que j'avais pris l'habitude de dire comme tout le monde : "Bon, tous ces programmes simples et autres : ils peuvent faire tout ça, mais ils ne vont pas pouvoir faire ça (et encore ça)." Ainsi par exemple, avant que je travaille sur le livre de NKS, une des choses que je croyais était que je pouvais faire des programmes simples représentant les phénomènes standard de la physique – pas au niveau de la physique fondamentale, mais au niveau de choses comme la façon dont les flocons de neige se forment, des trucs comme ça – je croyais par exemple que des systèmes biologiques avec adaptation et sélection naturelle devait finir par conduire à des niveaux de complexité si hauts qu'ils seraient impossible à saisir par les très simples programmes auxquels je m'intéressais. Mais j'ai pris conscience, à force de travailler sur le livre de NKS et autres, que c'était carrément faux. C'est un tout autre débat, mais je croyais en gros qu'il y avait différents niveaux de choses, pouvant ou ne pouvant pas être réalisées par des programmes simples. Et à force de travailler dans de plus en plus de domaines, je suis devenu de plus en plus convaincu que la richesse aisément disponible dans l'univers computationnel est assez grande pour que ça ne paraisse pas trop fou de penser que notre univers réel puisse être constitué de cette façon. Mais nous ne savons pas encore si c'est vrai tant qu'on n'aura pas mis la main sur la théorie finale, pour ainsi dire.

iii L'univers est-il un calculateur ?

GJ/HZ/GO : Maintenant nous avons un exercice linéaire très formel pour vous. En préambule, laissez-moi vous dire qu'au début de la conférence, il y a eu, je pense, de très intéressants échanges entre Tom Toffoli et Seth Lloyd, qui est un autre vieil ami de Santa Fe. Et c'est drôle qu'il ait dit qu'il avait fallu qu'il parte pour aider ses enfants à sculpter des lanternes à feux follets, car peu de temps après, j'avais rencontré Seth m'invitant chez lui à Santa Fe, dans la maison qu'il avait loué, et qu'il y avait aussi sculpté de ces lanternes – c'était avant qu'il soit marié et ait des enfants. Notre autre

sculpteur de lanternes à feux follets était Murray Gell-Mann. J'écrivais une biographie de Murray à l'époque et il n'avait pas voulu coopérer, faisant même une sorte d'obstruction, mais assis là avec un prix Nobel sculptant des lanternes a aidé à briser la glace.

Bien sûr Seth a décrit sa grande vision de l'univers comme calculateur quantique – l'univers est un computer quantique, dit-il, puis le Dr. Toffoli a émis une objection intéressante et subtile et, à la base, je pense que vous avez dit qu'il y avait cette idée de relation computationnelle entre l'observateur et le système, que quelqu'un doit pouvoir l'observer. Plus tard – pour passer à un autre contexte de la même session – quand Seth a fait une sorte de blague sur les élections et vous avez dit citant Staline, peu importe qui vote, c'est ceux qui comptent les votes qui ont de l'importance, ce genre de chose m'a fait repenser à ça, à toute cette idée d'endo-physique sur l'observation de l'univers à partir de l'intérieur ou de l'extérieur. Donc je me demande : est-ce que la vision de Seth de l'univers non pas comme un computer, mais disant qu'il est un computer, implique un point de vue divin de quelqu'un observant la computation ? Qu'est-ce que l'univers calcule, s'il calcule ? L'univers est-il un computer ou est-il comme un computer – ou est-ce exactement la même chose ?

Stephen Wolfram : Je pense à quelques trucs à dire, assez évidents. Il y a la **Question** : "Qu'est-ce qu'un modèle ?" En d'autres mots si vous voulez, dans le cadre de la physique traditionnelle : "Est-ce que le mouvement de la Terre est une équation différentielle ?" Non, ce n'en est pas une – il est décrit par une équation différentielle, mais ce n'est pas une. Et je ne pense pas, peut-être que d'autres gens voient les choses différemment, mais ma vision de la computation comme quelque chose qui sous-tend la physique est que la computation est une description de ce qui se passe en physique. Il n'y a pas vraiment de raisons de dire que l'univers est un computer. C'est simplement quelque chose qui peut se décrire comme un processus computationnel, ou quelque chose qui opère selon les règles d'un certain programme.

Cris Calude : Je suis d'accord et j'ajouterai une idée. Quand vous avez un phénomène réel, et un modèle – un modèle, habituellement, est une version simplifiée de la réalité. Pour pouvoir juger si ce modèle est utile ou non, il faut obtenir quelques résultats (en utilisant le modèle). Donc je jugerai le mérite de cette idée de l'univers modélisable par un gigantesque calculateur, en disant d'abord : " Citez-moi s'il vous plaît trois faits importants que ce modèle révèle au sujet de l'univers, et que les modèles classiques laissent dans l'ombre."

Ed Fredkin : Je vais vous en citer un dans une minute. Ce qui se passe

avec un processus computationnel, c'est que normalement on pense à un groupement de bits qui évoluent dans le temps, plus un moteur – le computer. La chose intéressante avec les processus informationnels (digitaux), c'est qu'ils sont indépendants du moteur, ceci dû à la nature du processus : tout moteur qui est universel (et c'est difficile d'en fabriquer qui ne le soient pas) – tant qu'il a suffisamment de mémoire – produira exactement le même processus informationnel que n'importe quel autre. Donc, si on pense que la physique est un processus informationnel, alors on n'a pas à se soucier de la forme du moteur – car le moteur n'est pas présent. Autrement dit, si l'univers est un processus informationnel, alors le moteur, s'il y en a un, est ailleurs.

Greg Chaitin : Georges, vous posez une question qui est une question philosophique à la base. C'est l'épistémologie versus l'ontologie. En d'autres mots, quand vous dites que l'univers ressemble à un computer, que c'est une modélisation pouvant être utile – c'est un point de vue épistémologique, qui nous aide à comprendre, qui nous fournit une certaine connaissance. Mais une question plus profonde portera sur ce qu'est réellement l'univers. Pas simplement avoir un petit modèle, avec lequel on va se débrouiller de mieux comprendre, d'apprendre des choses. Donc c'est une question bien plus ambitieuse ! Les anciens Grecs, les présocratiques, avaient de très belles idées ontologiques : le monde est nombre, le monde est ceci, le monde est cela. Et la physique fondamentale désire aussi répondre à des questions ontologiques : "de quoi le monde est-il réellement fait à un niveau fondamental ?" Ainsi, il est vrai que nous travaillons très souvent modestement sur des modèles, mais quand on commence à examiner la physique fondamentale et à fabriquer des modèles pour elle, et si un modèle fonctionne particulièrement bien – on commence à penser que le modèle est la réalité. Que c'est réellement une étape ontologique de plus. Et je pense que la philosophie moderne ne croit pas à la métaphysique, et certainement pas à l'ontologie. Ce n'est plus à la mode. Ils se contentent juste de considérer les questions épistémologiques ou de langage, mais nous, mathématiciens et physiciens, nous continuons à nous occuper de la question ontologique dure : Si vous faites de la physique fondamentale, vous cherchez la réalité ultime, vous travaillez sur l'ontologie. Et de nombreux travaux de beaucoup de physiciens actuels ressemblent vraiment à de la métaphysique – quand vous vous mettez à parler de tous les mondes possibles, comme Max Tegmark le fait, ou bien d'autres, ou David Deutsch sur cet exemple. Ainsi, les philosophes sont devenus très timides, mais certains d'entre nous ici, nous continuons dans la tradition des présocratiques.

Ed Fredkin : Le problème – un des problèmes – que nous avons est le problème cosmogonique de ce qui est à l'origine de l'univers. On a deux types

de faits contradictoires : d'une part, on a un tas de lois de conservation, en particulier l'énergie massique est conservée, et d'autre part on observe que l'univers a un début, semble-t-il, avec un big-bang assez récent à vrai dire, juste treize ou quatorze milliards d'années en arrière. Il y a une contradiction de base dans ces deux déclarations : car si quelque chose possède un début et que vous avez une loi de conservation, d'où viennent toutes les choses, ou comment c'est arrivé là ? Et l'idée d'un début à un certain moment est aussi problématique – les lois de la physique ne peuvent pas nous aider sur ce point-là parce que, si on réfléchit bien à tous ces détails, ce n'est pas tant pourquoi la matière est soudainement apparue, mais pourquoi y a-t-il une physique et pourquoi cette physique-là, et tout un tas de trucs comme ça.

On peut lever la main et donner une sorte de réponse pas très satisfaisante. Qui serait qu'il faut imaginer l'existence d'un certain lieu autre – que j'appelle simplement 'autre' – et là, dans cet autre lieu, pour une raison quelconque, il y a un moteur, et ce moteur est le siège d'un processus informationnel. On peut certainement arriver à certaines conclusions faibles quant à cet autre lieu, car on peut établir certains nombres quantifiant la taille que ce computer doit avoir. Autrement dit, si on se pose la question de ce qu'il faudrait pour faire tourner un computer émulant notre univers – on peut trouver certains nombres et quantités, s'il est possible de comprendre seulement quelques points, et ainsi faire un petit nombre de déclarations [suppositions cultivées] au sujet de cet autre type de lieu. L'essentiel quant à cet 'autre' est que c'est un lieu qui n'a pas besoin de posséder de lois de conservation, un lieu qui n'a pas besoin d'avoir de concepts tel que 'commencement' et 'fin' – il n'y a pas toutes ces nombreuses contraintes. Et une des merveilleuses choses concernant la computation, c'est que c'est un des concepts les moins exigeants, quand on dit : d'accord, que nous faut-il pour avoir un moteur computationnel ? Eh bien on a besoin d'une sorte d'espace. Quel type d'espace ? Combien de dimensions doit-il avoir ? Bon, ça pourrait être trois, ça pourrait être deux, ou une, ou sept – ça n'a pas d'importance. Toutes peuvent réaliser les mêmes computations. Bien sûr, si vous avez un espace unidimensionnel, vous pouvez avoir besoin de beaucoup de temps pour surmonter cet handicap. mais a-t-on besoin des lois de la physique telles que nous les connaissons ? Non, pas besoin des lois de la physique telles qu'on les connaît. En fait, les besoins sont si minimum pour avoir une computation, en comparaison avec la richesse de la physique que nous avons, que c'en est très très simple. Je me souviens d'une nouvelle de science-fiction, écrite par un auteur polonais, dans laquelle il y a un robot qui peut faire tout ce qui commence par la lettre 'n', et en polonais, comme en russe et en anglais, le mot 'nothing' (rien, en français) commence par un 'n', donc quelqu'un qui

s'ennuie dit : "OK, ne fait rien". Le robot se met au travail, et là où le ciel avait été blanc parsemé d'étoiles et de galaxies quelques minutes auparavant, il commence à s'éteindre doucement, petit à petit, galaxie après galaxie. Je l'admets, c'est juste une nouvelle de science-fiction, mais l'important c'est qu'on pourrait même aller jusqu'à se demander ce que seraient les motivations pour créer une émulation telle que celle-ci ? On peut imaginer qu'il existe une certaine question, pour ensuite avoir à y réfléchir : quelle serait cette question ?

On peut spéculer là-dessus. Mais l'essentiel, c'est qu'il y a une explication disant : bon, on a cette chose appelée 'autre' dont on ne sait rien – qui s'oppose à toutes les autres explications impliquant l'apparition d'un certain type de magie. Bah, je n'aime pas la magie, pour ma part.

Rob de Ruyter : Quelques mots selon une perspective biologique : Partons d'un point de vue naïf qu'il existe un monde extérieur et qu'il y a un cerveau, que ce cerveau a besoin de comprendre ce qui se passe dans le monde, et malheureusement, peut-être – ou heureusement – ce cerveau est un moteur vraiment bien adapté pour traiter de l'information dans la savane, ou dans les arbres, ou n'importe où ailleurs. Je ne pense pas que le cerveau lui-même soit un moteur computationnel universel – du moins, je ne le vois pas de cette façon – mais c'est un appareillage extrêmement bien adapté pour traiter l'information qui vient à nos organes des sens, tirée du monde dans lequel il se trouve qu'on habite. Donc si on veut commencer à réfléchir à des choses plus complexes ou plus profondes, il faut développer des outils nous permettant de traduire nos pensées au sujet des phénomènes observés dans le monde. D'un autre côté, tout comme on façonne un marteau ou une pince, en interagissant avec les phénomènes présents dans le monde, on a besoin de développer des outils pour réfléchir [à eux]. Je n'ai aucune idée des limitations liées à la structure de notre cerveau, et de ce qu'elles imposent – je pense qu'il doit y avoir des limitations (dans notre façon de penser) qui imposent une structure à ces outils développés pour nous aider à réfléchir sur les choses. Et je pense que la computation, d'une certaine manière, est l'un des outils que nous essayons de développer pour traiter avec le monde qui nous entoure. Et ce que la computation nous permet de faire, comme par les mathématiques, c'est d'être capable de développer de longues chaînes de raisonnement dont on ne se sert pas normalement, mais qu'on peut rendre raisonnables sur de très longues séries, des séquences d'observations compliquées sur des phénomènes présents dans le monde qui nous entoure. Ainsi, ce qui m'intéresse, c'est cette relation entre la façon dont nous pensons – et la façon dont nous avons évolué pour penser le monde autour de nous – et les choses auxquelles nous pensons aujourd'hui en termes d'observations scientifiques et d'origine

de l'univers. Jusqu'à quel point le hardware que nous transportons, portons en nous, nous informe, et détermine les types d'outils, de computations et de stratégie que nous utilisons ?

Tom Toffoli : J'aimerais vous donner quelque exemples pour illustrer pourquoi la question de savoir si l'univers est un computer, est vraiment une question difficile. En un sens, cela ressemble à la question de ce qu'est la vie. Prenons par exemple le concept d'aléatoire : disons que vous achetez un générateur de nombres aléatoires et commencez à produire des nombres avec. Le premier nombre que vous obtenez est le 13, puis vous avez 10, 17 et ainsi de suite – puis vous commencez à vous poser la question sur les nombres obtenus : quel est leur degré d'aléatoire ? Qu'est-ce que l'aléatoire ? Existe-t-il un nombre aléatoire ? 13 est-il un nombre aléatoire ? Qu'en est-il de 199999 – est-ce un nombre aléatoire ? Puis vous réalisez évidemment que l'aléatoire n'est pas une propriété du nombre, c'est une propriété du processus. Vous vous payez un générateur de nombres aléatoires car vous voulez être surpris. Vous ne voulez pas savoir ce qui va sortir. Si vous le saviez – vous ne le considèreriez pas comme aléatoire, car ce serait parfaitement prévisible. Donc on utilise ce terme de 'nombre aléatoire' comme abréviation pour toute séquence produite par un générateur de nombres aléatoires. Je vais vous donner un autre exemple : quelqu'un a essayé de poser un brevet sur les icônes. Quand elles furent inventées, les icônes étaient réduites : 16 par 16 pixels en noir et blanc, et on pouvait dessiner quelques images simples avec ces 16 par 16 bits. On peut vouloir déposer un brevet là-dessus. Et un averse homme d'affaires quelconque s'est dit : voyons voir si je vais devant un juge et que j'essaie de poser un brevet sur l'ensemble complet des icônes 16 par 16 pixels, toutes les icônes, et tout le monde devra me payer. Maintenant si vous êtes le juge et devez décider si vous pouvez ou non permettre à quelqu'un de déposer un brevet non pas juste sur une icône, mais sur toutes les icônes pouvant être faites de ce type. Donc si vous êtes le juge, vous avez deux façons d'agir : ou vous dites (a) qu'il faut me donner une bonne raison pour laquelle cette icône possède vraiment quelque chose d'intéressant, ou (b) il faut payer 5 cents pour chacune des icônes que vous enregistrez, et étant donné qu'il y en a 2256, pas besoin d'exercer aucun jugement, toute l'affaire tourne à la grosse contribution à la communauté.

Autrement dit, la question est : qu'est-ce qui fait qu'une icône est ce qu'elle est ? C'est juste le fait qu'elle soit 16 par 16 bits, ou alors on a une raison de croire qu'il y a quelque chose d'utile en elle ? Brian Hayes, dont vous parliez il y a un instant, a dit un jour : "Ce qui me surprend, c'est que la plupart des gens ne se servent pas d'un ordinateur pour ce qui le rend unique et puissant – en l'occurrence que c'est une machine programmable." Ma défi-

nition partielle d'un computer est : quelque chose qui peut calculer (évaluer) beaucoup de fonctions différentes. S'il ne peut évaluer qu'une fonction, alors je ne l'appellerais pas un computer, mais plutôt une machine spécialement dédiée ou quelque chose comme ça. Donc, si on trouve les formules qui nous fassent considérer l'univers comme un computer, il se pourrait qu'on soit surpris car alors justement, l'univers lui-même devient une machine spécialement dédiée. Je veux dire, si on connaît les formules, on connaît la machine, on connaît les conditions initiales, et ça avance : tick, tick tick, tick. C'est le plus gros computer mais personne ne peut le programmer car on est à l'intérieur.

Donc la définition d'un computer est un peu comme la définition de la vie, la définition de l'évolution ou de l'adaptation : s'il n'existe pas de composante d'adaptativité, je n'appellerais pas ça un computer. Maintenant cela pourrait être mieux formalisé, je parle d'une façon intuitive, mais je me pose certaines de ces questions pour essayer de clarifier un peu si c'est vraiment ce qu'on cherche.

Greg Chaitin : Voyons, j'aimerais attaquer là-dessus – le meilleur moyen pour moi de penser sur quelque chose, c'est de faire des affirmations qui soient beaucoup trop fortes (au moins, les idées sortent). Donc il faut que l'univers soit un computer, comme Stephen le disait, car le seul moyen de comprendre quelque chose, c'est de le programmer. Moi-même, je me sers du même paradigme. Chaque fois que j'essaie de comprendre quelque chose, ma façon de procéder consiste à écrire un programme informatique. Donc le seul modèle fonctionnel possible de l'univers est nécessairement d'être un computer – un modèle computationnel. Je dis modèle fonctionnel car c'est le seul moyen qu'on a de comprendre quelque chose : en écrivant un programme, en le faisant fonctionner et en le débbugant. Puis essayer de faire tourner ça sur des exemples et autres. Pour ainsi dire, comprendre quelque chose seulement si on peut le programmer. Maintenant que se passe-t-il si l'univers décide qu'il n'est pas – qu'on ne peut pas en faire un modèle computationnel. Bien : pas de problème. Ça veut juste dire qu'on utilise les mauvais computers. Ainsi, si cet univers est plus puissant que ce qu'un modèle de computer peut l'être, ça signifie que notre notion de ce qu'est un computer est fausse, et on a juste besoin d'une notion qui soit plus puissante, et alors les choses seront en synchronisation. D'ailleurs, il y a moyen de définir l'aléatoire de nombres individuels fondés sur un computer [s'entend : des nombres infinis, selon Toffoli], mais bon, c'est un autre problème.

Stephen Wolfram : Une remarque à faire, en relation avec l'usage de la computation comme modèle d'un univers : on se sert de quelque chose de bien particulier quand on fait de la modélisation, on se sert de modèles en

tant qu'idéalisation de ce qui se passe. On dit qu'on va se procurer un modèle de flocon de neige, ou de cerveau, ou autre, sans imaginer qu'on va avoir un modèle parfait ! C'est un cas très inhabituel que de s'occuper à modéliser la physique fondamentale (peut-être que modéliser n'est pas le terme juste, car on imagine alors avoir vraiment un modèle précis qui reproduira notre univers réel dans tous ses détails). Ce n'est pas le même genre de choses que ce qui a été la tradition de la modélisation en science naturelle, c'est bien plus. Donc quand vous dites, quand vous parlez de ce qui le fait marcher et autre, ça ressemble beaucoup plus à une discussion mathématique : on ne se poserait pas la question si on réfléchissait à un résultat mathématique, si on travaillait sur certains résultats de la théorie des nombres, par exemple, on ne se poserait pas tout le temps la question "Bon, qu'est-ce qui fait marcher tous ces nombres ?" C'est juste pas la question qui vient quand on s'occupe de quelque chose avec un modèle exact pour les choses.

Une autre remarque à faire sur la question de jusqu'à quel point nos efforts de modélisation sont liés à la capacité de nos cerveaux pour le faire et autre. Une des choses qui excite ma curiosité est : s'il s'avère qu'on peut trouver une représentation précise, une nouvelle représentation (meilleur mot que modèle) pour notre univers en terme de programmes simples et qu'on trouve que c'est tel et tel numéro de programme – qu'en conclure alors ? C'est une drôle de situation scientifique, ça rappelle un peu quelques situations scientifiques passées, comme par exemple Newton qui parlait du fonctionnement des orbites planétaires et autre, en déclarant qu'une fois les planètes sur leur orbite, on pouvait se servir des lois de la gravité et autre pour comprendre ce qui arrive – mais il ne pouvait pas imaginer ce qui avait bien pu installer les planètes au départ dans le mouvement de leur orbite. Donc il disait "Bon, la Main de Dieu doit avoir mis les planètes en mouvement au départ." Et c'est pareil avec les théories de Darwin : une fois la vie apparue, alors la sélection naturelle va inexorablement nous mener à toutes les choses qu'on voit en biologie. Mais comment provoquer la vie au départ – il ne pouvait pas l'imaginer. Donc certaines des choses excitant ma curiosité seraient : si en fait on arrive à sortir une représentation exacte de l'univers en tant que programme simple – que fait-on alors et pouvons-nous imaginer quel genre de conclusion on peut tirer, sur pourquoi ce programme-là et pas un autre, et ainsi de suite ? Ainsi, l'une des possibilités serait qu'on trouve que c'est le programme numéro 1074 ou un autre, peu importe. Le fait que ce soit un nombre si petit pourrait être une conséquence du fait que nos cerveaux sont organisés comme ils le sont, car ils sont fabriqués à partir de cet univers de telle façon que ce soit inévitable, et que dans toutes les énumérations qu'on pourrait utiliser, notre univers s'avérerait être un univers doté d'un petit

nombre. Je ne pense pas que ce soit le cas – mais c’est une de ces prophéties auto-suffisantes : parce qu’on existe dans notre univers, notre univers devra avoir des lois qui nous sembleront simples et intuitives, d’une façon ou une autre. Je pense que c’est plus clair que ça, mais c’est une des résolutions potentielles de cette **Question** : si maintenant on a notre représentation de l’univers, que conclure métaphysiquement du fait que c’est cette représentation particulière de l’univers et pas une autre.

Tony Leggett : Je considère avec respect la forte et puissante déclaration pouvant se rattacher à l’argument de Seth Lloyd, notamment que l’univers est un computer, j’ai juste une petite question très simple et très naïve : il me semble que si on a exploré cette déclaration, la déclaration contraire ne l’a pas été. Donc ma question à Seth Lloyd est : "A quoi ressemblerait un univers qui ne soit pas un computer ?" Et j’ai bien peur, pour l’instant, de ne pas avoir trouvé cette déclaration particulière d’une grande utilité. Je pense que la thèse disant qu’il pourrait être utile de voir l’univers à travers le cadre particulier de la science computationnelle en posant des questions différentes sur lui, pourrait être d’une certaine aide, quoique je dois dire que je ne suis toujours pas convaincu que le voir de cette façon puisse nous aider à répondre à certaines questions très évidentes, certaines d’entre elles étant habituellement associées à un certain niveau de controverse, à savoir est-ce que le principe anthropique¹⁷ possède une signification en physique ? C’est-à-dire pourquoi les constantes naturelles telles que nous les connaissons ont les valeurs particulières qu’elles manifestent ? C’est peut-être pour une certaine raison arbitraire, ou peut-être une certaine raison plus profonde. Maintenant, bien sûr, il y a ici pléthore d’arguments et spéculations sur toutes sortes de choses, mais autant que je puisse le voir, ils ne semblent pas donner lieu à une relation directe (entre l’univers et un modèle computationnel spécifique, ou entre l’univers et un computer). Mais j’aimerais vraiment entendre un

17. Le *principe anthropique* en cosmologie, possède une longue et vénérable histoire – en fait il remonte bien avant la naissance de la physique telle que nous la connaissons. Les versions plus modernes partent de deux observations globales. La première est que, dans le formalisme courant de la physique des particules et de la cosmologie, il y a un grand nombre de constantes qui ne sont pas déterminées par la théorie elle-même, mais qu’on doit installer "à la main". La seconde observation est que les conditions physiques nécessaires pour l’existence de la vie – plus encore pour son développement jusqu’au stade humain – sont extrêmement rigoureuses. Il est facile d’en arriver à la conclusion que, pour qu’il existe un type quelconque d’êtres conscients, les constantes de base de la nature doivent être exactement ce qu’elles sont, ou au moins extrêmement proches de ce qu’elles sont. Le principe anthropique inverse alors cette déclaration en disant que la raison pour laquelle les constantes fondamentales possèdent les valeurs qu’elles ont effectivement, c’est qu’autrement il n’y aurait personne pour s’en émerveiller". (The Problems of Physics, par Anthony J. Leggett, 1987 Oxford University Press, pp. 145-146)

argument plausible sur pourquoi ce point de vue nous ferait progresser sur les types de questions que je viens de mentionner.

Ed Fredkin : Je voudrais réagir un peu à ce qu'a dit Stephen. Il existe des domaines où on utilise les computers pour écrire des programmes qui sont des modèles exacts, parfaits et exacts dans tous les sens possibles – c'est quand on conçoit un programme pour émuler un autre ordinateur. On le fait tout le temps pour l'écriture de la carte retraçant le programme et pour le débogage, quand vous écrivez un émulateur pour le computer sur lequel le logiciel est exécuté. Ou quand vous voulez exécuter un logiciel qui est fait pour un autre ordinateur, comme le Mac quand il passait sur CPU du 68000 au PowerPC : ils avaient conçu un émulateur qui est l'exact implémenteur du logiciel d'un autre ordinateur.

C'est lié à quelque chose qu'on a l'habitude d'appeler 'la Tyrannie de l'Universalité'. En l'occurrence : "Impossible de jamais connaître le plan du computer qui exécute la physique car tout computer universel peut le faire aussi." Autrement dit, s'il existe un computer digital exécutant toute la physique, alors évidemment tout computer peut le faire, mais ensuite, après m'être convaincu du bien-fondé de cette position, j'en suis venu à une perspective différente : que si le processus exécutant la physique est digital et que c'est une sorte d'AC, il va y avoir alors un modèle qui va copier son fonctionnement pas à pas. Et ce sera probablement possible de trouver le plus simple de ce genre de modèle, de sorte que si un certain type d'évidence expérimentale nous montrait que la physique est une espèce de processus discret et digital ressemblant à un AC, je crois alors qu'on serait capable de trouver le processus exact (ou, si vous voulez, l'un des ensembles réduits de ces processus) qui l'implémente exactement.

iv L'univers est-il discret ou continu ?

GJ/HZ/GO : Peut-être un bon moyen de revenir à la question soulevée par Tony Leggett, finalement c'est la même question que posait notre hôte Adrian German après que nous ayons visionné ce petit film d'introduction ce matin, notamment : y a-t-il une différence fondamentale entre une physique computationnelle, ou un modèle computationnel, ou une émulation de l'univers, et la mécanique quantique ? Ou bien existe-t-il une distinction fondamentale entre une physique discrète et une physique continue ? Quelqu'un veut-il réagir là-dessus ?

Stephen Wolfram : (s'adressant à Georges Johnson) : Si la question

porte sur l'existence d'une sorte de test définitif pour savoir si l'univers est discret ou fondamentalement continu...

GJ/HZ/GO : Oui, s'il existe un conflit pouvant se situer à un niveau plus profond des deux côtés.

Stephen Wolfram : Si c'est ça la question – par exemple, dans les types de modèles que je me suis efforcé d'étudier, il y a tellement de façons différentes de formuler ces modèles que la question "est-ce discret ou continu ?" devient un peu bizarre. Je pense qu'on pourrait dire : on va représenter ça sous une certaine forme algébrique par laquelle on va avoir l'air de parler de ces objets très continus – et peu importe si ça tourne au discret, ça peut bien être une représentation discrète (qui sera beaucoup plus facile à traiter). Je pense donc qu'au niveau des modèles que je considère plausibles, la distinction entre continuité et discrétion est beaucoup moins claire que prévue. Je pense que si on se pose cette question, on va finir par se poser des questions très 'non-physiques' comme, par exemple, combien d'information peut tenir en principe dans ce volume d'espace. Je ne suis pas sûr que sans opérationnaliser cette question, elle soit d'un grand intérêt ou signification.

Tom Toffoli : Je voudrais dire quelque chose qui sera très bref. Regardez les automates cellulaires, ils semblent être un paradigme pour la discrétion. mais il se trouve que l'une des caractérisations des AC est qu'ils sont un système dynamique [qui réalisent certains types de translations] et qu'ils sont continus par rapport à une certaine topologie semblable à la topologie de l'ensemble de Cantor, continus exactement au sens même de la définition de continuité dans le calcul, celui qui est étudié en première année. Mais la chose intéressante, c'est qu'il s'agit de la topologie de l'ensemble de Cantor, inventée par Cantor (celle avec l'intervalle où vous retirez le tiers central etc.) Et le fait est que cette topologie pour ACs – pas au sens d'une topologie géométrique, mais au sens d'une topologie ensembliste de circuits dotés de portes ayant un nombre fini d'entrées et un nombre fini de résultats en sortie – c'est exactement la topologie de Cantor, donc c'est une sorte de topologie universelle pour la computation. Et on revient donc au point de départ, notamment (a) que quelque chose qui ne fut pas inventé par Cantor pour décrire les ordinateurs, représente en fait la topologie naturelle pour décrire les computers discrets et (b) dès l'instant où on prend un treillis in[dé]fini, alors on obtient de la continuité, exactement le même type de continuité, continuité d'état, que celle de la dynamique obtenue lorsqu'on étudie les fonctions continues. Voilà quelques-unes des surprises auxquelles on arrive en travaillant sur les choses.

Stephen Wolfram : Je voudrais juste dire quelque chose par rapport

à la question de comment peut-on dire de telles choses, vous savez, ce que Tony disait : "Comment peut-on dire que tout ça n'est pas complètement absurde?" C'est ça ?

Tony Leggett : Réfutation de la déclaration négative.

Stephen Wolfram : Oui, d'accord. Nous allons seulement être vraiment sûr de savoir si et quand on arrivera à obtenir une théorie, une représentation qui soit l'univers, et qui puisse être sous une forme computationnelle commode à représenter. Alors les gens diront : "Super ! Cette idée de computation est juste, c'était évident depuis longtemps, tout le monde y pensait depuis des lustres ..."

Tom Toffoli : A tel point qu'ils diront sûrement : "En fait, c'est trivial !"

Stephen Wolfram : Et je pense que tant qu'on n'en sera pas arrivé là, on peut argumenter pour ou contre à l'infini sur ce qui est plausible ou pas, et ce sera toujours difficile de décider sur quoi c'est basé. D'ailleurs, en science, ce genre de chose tend à se décider d'une manière étonnamment sociologique. Par exemple le fait que les gens puissent imaginer sérieusement que les orientations de la théorie des cordes soient considérées comme des moyens de modéliser la réalité d'un univers physique – c'est intéressant et c'est de la belle mathématique – mais c'est un phénomène sociologique qui fait [ou qui oblige] à ce qu'on grève sérieusement les dépenses pour d'autres approches. Et ce sont des causes historiques qui font que l'approche dont nous nous servons (les idées de computation) n'est pas le thème dominant dans la réflexion sur la physique au jour d'aujourd'hui. Je pense que c'est purement une question d'histoire. Il se pourrait bien qu'à la place de la théorie des cordes, les gens soient en train d'étudier toutes sortes de bizarres systèmes d'ACs ou de réseaux, ou n'importe quoi d'autre, et que se soit tissé le même type de toile mathématique élaborée que ce qui a été fait autour de la théorie des cordes, avec des gens tout aussi convaincus que le sont les théoriciens des cordes qu'ils sont sur la bonne voie. Je pense qu'à ce stade, jusqu'à ce qu'on ait une réponse définitive, on n'en sait simplement pas assez pour être capable de dire quelque chose avec certitude, et c'est vraiment une question purement sociologique que de pouvoir dire si on est dans la bonne direction ou non. L'IA ressemble tout-à-fait à ce genre de chose, les gens vont débattre sans fin pour savoir s'il est possible d'obtenir une IA et ainsi de suite – et certains d'entre nous sont en train de produire de gros efforts pour essayer de faire des choses pratiques qui puissent être identifiées comme pertinentes par rapport à ça. Je pense qu'en fait la question de l'IA est plus difficile à trancher que la question de la physique. Parce qu'en physique, une fois qu'il sera probable (d'après moi) qu'on peut montrer que la représentation

en question est évidemment celle de l'univers réel, le débat sera clos. Alors que la question de l'IA sera plus difficile à fermer.

Cris Calude : Apparemment, il y a un antagonisme entre la vision discrète et continue. Mais si on regarde les mathématiques, il existe des univers mathématiques où discret et continu coexistent. Bien sûr, ce que Tom disait, l'espace de Cantor est un exemple très intéressant, mais il se pourrait qu'il soit trop simple pour le problème dont nous parlons. Par exemple, l'analyse non-standard est un autre univers où on trouve ce même phénomène, vous avez de la discrétion et de la continuité, et peut-être, dans la mesure où les mathématiques peuvent dire quelque chose de l'univers physique, il se pourrait qu'on ait affaire à un mélange de continuité et de discrétion, certains phénomènes pouvant se révéler à travers la discrétion et certains autres le seront par la continuité et les fonctions continues.

Greg Chaitin : Encore une fois, je vais exagérer – à bon escient. Je pense que la question se pose ainsi : discrétion versus continuité. Et je vais vous dire pourquoi je suis du côté de la discrétion.

La raison en est : je désire que le monde soit compréhensible ! Maintenant, il existe différentes façons de dire cela. L'une d'elles serait : Dieu n'aurait pas créé un monde que nous ne pourrions pas comprendre. Ou toute chose arrive pour une bonne raison (le principe de la raison suffisante). Et d'autres façons. Je suppose qu'on peut me qualifier de néo-pythagoricien car je pense que le monde est plus beau, s'il est compréhensible. Nous sommes des penseurs, nous sommes des rationalistes – pas des mystiques. Un mystique est quelqu'un qui entre en communion avec un monde incompréhensible, ressent une sorte de communauté, et est capable d'entrer en relation avec elle. Mais nous voulons rationnellement comprendre, donc le meilleur univers est celui qu'on peut complètement comprendre, et si l'univers est discret, on peut le comprendre – il me semble. C'est quelque chose que vous avez dit, à un certain moment, Ed – c'est absolument et totalement compréhensible car on lance le modèle et le modèle est exactement ce qui se passe.

Maintenant, un univers qui se sert de la continuité est un univers où aucune équation n'est exacte, c'est ça ? Car on dispose seulement d'approximations jusqu'à un certain ordre. Donc je dirai aussi : un univers serait plus beau s'il était discret ! Et quoiqu'on en soit arrivé maintenant à l'esthétique, ce qui est encore plus compliqué, je continuerai à dire qu'un univers discret est plus beau, une plus grande œuvre d'art à créer pour Dieu – et je ne suis pas religieux, par ailleurs. Mais je pense que c'est une très bonne métaphore – ou peut-être suis-je religieux d'une certaine manière, qui sait ?

Une autre façon de le présenter c'est disons que cet univers possède de

la continuité et une précision infiniment embrouillée et tout ça – en bref, ça ne va pas. Pourquoi Dieu n'aurait-il pas créé un univers aussi beau que possible ?

Tom Toffoli : Il aurait du s'adresser à vous, Greg !

Greg Chaitin : Quoi ? ... Non .. Non ... je pense que là-dessus, peut-être Stephen est le meilleur candidat pour proposer un ...[tout le monde rit encore, dont Chaitin qui continue] ... il me semble que ce serait plus beau. Vous voyez, ce serait plus compréhensible, plus rationnel, ça montrerait la puissance de la raison. maintenant peut-être que la raison est dans l'erreur quand elle postule que l'univers est compréhensible – que ce soit comme un postulat fondamental ou parce Dieu est parfait et bon, et n'aurait pas voulu créer un univers autre, si on veut reprendre un point de vue théologique ancien. C'est peut-être tout faux, mais c'est une des raisons pour laquelle je suis un néo-pythagoricien, par ce que je pense que ce serait un univers plus beau , ou plus compréhensible.

Stephen Wolfram : J'ai un point de vue plus pragmatique, qui est que si l'univers est quelque chose qui peut se représenter par quelque chose comme un programme simple discret, alors il est réaliste de croire qu'on peut le trouver en le cherchant. Et ce serait embarrassant si l'univers se trouvait en effet au-delà du premier milliard d'univers, disons, qu'on peut trouver par énumération, et on n'aurait jamais la patience d'aller le chercher. [Réaction soutenue de la part des autres membres de la table, tout particulièrement Greg Chaitin, qu'on entend rire.] Il est possible que l'univers ne soit pas accessible de cette façon – mais ce n'est pas encore exclu ! On en est là, pour l'instant. peut-être dans 10-20-50 ans on pourra dire : oui, on a observé tout le premier [milliard] je sais pas combien – ce sera comme rechercher des contre-exemples de l'hypothèse de Riemann, ou quelque chose comme ça – et on pourra dire qu'on a observé le premier quadrillion des univers possibles et aucun d'eux n'est notre univers réel, donc on va commencer à perdre confiance dans le succès de cette approche. Mais pour l'instant, on n'en est même pas à l'étape de départ.

Ed Fredkin : Si c'était un univers à une dimension, Stephen [Wolfram], vous auriez déjà trouvé la règle, n'est-ce pas ? Car vous les avez toutes explorées ...

Tony Leggett : Bon, Georges, je pense avoir soulevé la question de savoir si la mécanique quantique a quelque chose à voir avec cette question, donc laissez-moi juste faire un petit commentaire là-dessus. Je pense que si on réfléchit à la structure générale de la mécanique quantique, et aux moyens par lesquels on vérifie ses prévisions, on en arrive à la conclusion que

presque toutes les expériences (et je vais prendre un risque et dire toutes les expériences) qui nous ont montré des choses intéressantes sur la mécanique quantique font de la mesure de variables discrètes, en fait. Les expériences sur ce qu'on appelle la cohérence quantique macroscopique, celles sur le théorème de Bell, et ainsi de suite – à la base, elles se servent toutes de variables discrètes en pratique. Maintenant, bien sûr, le formalisme de la mécanique quantique est un formalisme continu. On permet à des amplitudes d'avoir des valeurs arbitraires, mais on ne mesure jamais vraiment ces choses. Et je pense que tout ce qu'on peut dire quand on mesure parfois des choses comme la position et la vitesse qui sont [apparemment] des variables continues – à bien y regarder, vous verrez que le véritable montage opérationnel est tel qu'on mesure concrètement quelque chose de discret. Donc les mesures faites dans le cadre de la mécanique quantique, qui se réclament habituellement d'être faites sur des variables continues, sont le fait de variables discrètes. Ainsi, je pense que d'après le point de vue ontologique, posé comme tel, on peut dire que la mécanique quantique joue en faveur du point de vue discret.

v L'univers est-il aléatoires ?

GJ/HZ/GO : Quand Hector Zenil, Gerardo Ortiz et moi discussions sur les bonnes questions qui stimuleraient le débat, nous avons pensé que peut-être on devrait demander quelque chose qui soit vraiment vraiment à la base de l'aléatoire – et l'avantage d'être un journaliste, tout particulièrement un journaliste scientifique, c'est de pouvoir poser à des gens très intelligents des questions sur des sujets qui vous ont toujours rendu perplexe. Et il y a quelque chose qui m'a toujours intrigué – le SETI (Recherche d'une Intelligence Extraterrestre-Search for Extraterrestrial Intelligence) où on récupère ces signaux de l'espace qui sont ensuite analysés par des ordinateurs, à la fois par des superordinateurs et par des ordinateurs du SETI sur site, où vous faites donc d'un certain temps de travail du microprocesseur de votre PC, des ordinateurs en réseau etc. Ils recherchent une certaine structure dans ce qui paraît être un bruit aléatoire. Et je me demandais si l'apparence de pur bruit du signal obtenu ne l'était que jusqu'à un certain point – comme je pense que Tommaso Toffoli l'a suggéré – peut-être que l'aléatoire n'est que dans l'œil de l'observateur.

Si par exemple, quand on reçoit ce signal bruyant qui semble juste être des parasites – comment savoir si on n'est pas en train de recevoir les deux milliardièmes et cinquante septième décimales du développement de ? et leur suite ? Comment savoir si on n'est pas en train d'aligner la dix milliardième

de trilliardième étape et suivantes de la computation d'automates cellulaires de règle 30 ? Donc, je me demandais si vous pouviez m'aider là-dessus.

Tom Toffoli : Ceci n'est pas une réponse. C'est juste quelque chose pour exciter notre imagination. Supposons des gens prenant le calcul au sérieux et se disant : "Regardez, on n'utilise pas notre énergie efficacement car on laisse une certaine énergie – qui n'est pas complètement dégradée – partir ". Ainsi, ils commencent à améliorer les re-circulations, filtres et autres, et maintenant toutes les énergies thermiques sortantes sont aussi thermalisées que possibles. Car sinon, ils commettent un péché thermodynamique. Et c'est exactement ce qui se passe quand on regarde les étoiles. Elles émettent une certaine température tout près de l'équilibre thermique – ainsi on pourrait dire qu'il existe une évidence de prime abord qu'il y a des gens qui calculent et qui ne rejettent que du déchet, ils ne jettent rien qui ne soit encore recyclable ! Cela pourrait être une explication de pourquoi on voit toutes les étoiles avec ces températures-là !

Stephen Wolfram : Vous savez, j'ai réfléchi à la question sur le SETI et sur comment elle est liée aux types de choses dont nous parlons – je pense qu'elle nous amène vers des tas de choses intéressantes. J'ai été un grand fan du SETI, et comme je suis un gars pratique, j'ai pensé il y a des années à la possibilité de se servir des liaisons satellites inutilisées pour faire de la détection de signaux et autres. Et maintenant que j'ai travaillé sur le livre de NKS pendant un bon bout de temps, en réfléchissant au PCE, je ne suis plus du tout fan de SETI. Car j'ai réalisé que cette réflexion avait pour conséquence des déclarations comme "la météo possède un esprit propre". Il y a la question de ce que ça représenterait – quand on dit qu'on est à la recherche d'une intelligence extraterrestre – quelle est concrètement la version abstraite de l'intelligence ? C'est la même vieille question sur ce qu'est la vie, et pouvons-nous avoir une définition abstraite de la vie, qui serait séparée de notre expérience particulière de la vie sur Terre. Je veux dire que sur Terre, il est assez facile de dire si quelque chose est vivant, ou non. Car si c'est vivant, cela possède sûrement de l'ARN, avec certaines membranes, et toutes sortes de détails historiques qui font la connexion avec toutes les autres formes de vie que nous connaissons. Mais si vous dites, abstraitement : qu'est-ce que la vie ? La réponse n'est pas claire. A l'époque, dans l'Antiquité, c'étaient les choses qui se déplaçaient seules qui étaient vivantes. Plus tard, c'étaient les choses pouvant faire de la thermodynamique d'une façon différente, qui étaient vivantes. Mais on ne sait toujours pas – on ne dispose pas – d'une définition abstraite de la vie qui soit claire et séparée de toute histoire particulière. Je pense que c'est la même chose avec l'intelligence. La seule chose sur laquelle la plupart des gens (je crois) seraient d'accord, c'est que pour être intelligent, il faut

opérer une certaine computation. Et avec l'idée du Principe d'Equivalence Computationnelle, on peut dire qu'il y a beaucoup de choses ici-bas qui sont équivalentes quant au type de computations qu'elles peuvent opérer ...

Tom Toffoli : Mais vous pouvez aussi faire de la computation sans être intelligent !

Stephen Wolfram (répliquant à Toffoli) : C'est précisément la **Question** : est-ce qu'on peut – quelle est la différence, quelle est la caractéristique distinctive de l'intelligence ? Si on se tourne vers l'histoire, le tableau n'est pas clair du tout : un exemple fameux que j'aime bien se situe quand Marconi développait la radio et (il avait un yacht dont il se servait pour faire la navette sur l'Atlantique, et) à un certain moment il était au milieu de l'Atlantique, avec son mât radio monté sur le bateau – parce que c'était son business – et il a pu entendre ces drôles de sons, vous connaissez : ... wooou ...ooooeu ... wouuu ... ce genre de sons en plein milieu de l'Atlantique. Qu'est-ce que vous croyez qu'il en a conclu quant à l'origine de ces sons ? Il en a déduit que ça devait être des signaux radio venant des martiens ! Tesla était encore plus convaincu que c'était des signaux radio de Mars. Mais qu'est-ce que c'était en fait ? C'était des modulations de la ionosphère par rapport à la Terre, des processus physiques, quelque chose qui se produit dans le plasma. Donc la question était, comment distinguer la chose vraiment intelligente, s'il existe une quelconque notion de ça, de la chose qui est ... du truc computationnel qu'il est.

La même chose est arrivée avec les pulsars, quand les premiers pulsars furent découverts. Aux premiers temps de la découverte, il semblait que cette chose périodique de l'ordre de la milliseconde devait être une balise extraterrestre quelconque. Mais ça semblait trop simple. On pense maintenant que c'est trop simple pour être d'origine intelligente. Cette question est liée aussi au principe anthropique et à la question de savoir si notre univers est organisé d'une certaine façon uniquement pour qu'il soit capable d'héberger une intelligence comme la nôtre. Quand on se rend compte qu'il n'existe pas de définition abstraite de la réelle intelligence, on se retrouve juste (je pense) avec une histoire de computation à opérer. Alors l'espace des univers possibles supportant quelque chose comme l'intelligence devient beaucoup plus large, et on réalise en quelque sorte que cette idée de principe anthropique avec toute ses contraintes détaillées, n'a plus beaucoup de sens.

Il y a tellement de choses à dire là-dessus, que je vais laisser les autres parler.

GJ/HZ/GO : Aléatoire : c'est dans l'œil de l'observateur ?

Greg Chaitin : Georges, je suppose que ce serait un peu lâche de ma part de ne pas défendre la définition de l'aléatoire sur laquelle j'ai travaillé toute ma vie, mais je pense que c'est plus marrant de dire (je défends le rationalisme, comme vous savez) qu'un monde est plus compréhensible parce qu'il est discret, et pour cette raison il est plus beau. En fait, j'ai passé ma vie, ma vie professionnelle, à travailler sur une définition de l'aléatoire en essayant de la trouver, et je pense l'avoir trouvée, dans les mathématiques pures, ce qui est une drôle de place pour trouver quelque chose d'aléatoire. Quand vous dites que quelque chose est aléatoire, vous dites que vous ne pouvez pas le comprendre, d'accord ? Donc définir l'aléatoire correspond à l'esprit rationnel essayant de trouver ses propres limites, car donner une définition rationnelle de l'aléatoire est bizarre, c'est quelque chose de paradoxal que d'être capable de connaître ça, voyez-vous, être capable de définir l'aléatoire, ou être capable de savoir que quelque chose est aléatoire – car quelque chose est aléatoire quand il nous échappe ... je ne le formule pas bien, j'improvise, mais on a affaire à certains paradoxes là-dedans. La façon dont fonctionnent ces paradoxes, c'est que vous pouvez définir l'aléatoire mais vous ne pouvez pas savoir si quelque chose est aléatoire, car si vous le pouviez, ce ne serait plus aléatoire. L'aléatoire serait juste une propriété comme les autres, et on pourrait l'utiliser pour faire du classement. Et je pense qu'il y a une définition de l'aléatoire pour les nombres individuels sans prendre en compte le processus par lequel ces nombres viennent à nous : vous pouvez observer des chaînes individuelles de bits – des nombres en base 10 – et dire, au moins mathématiquement, ce que cela signifie pour eux d'être aléatoires. Maintenant, quoique la plupart de ces nombres en séquences soient aléatoires selon cette définition, le paradoxe est que vous ne pouvez jamais être sûr qu'un nombre individuel est aléatoire – donc je pense qu'il est possible de définir un concept d'aléatoire qui soit intrinsèque et structurel, et qui ne dépende pas du processus duquel vient le nombre, mais reste un problème avec cette définition, qui est : c'est inutilisable. Sauf à créer un paradoxe, ou à montrer les limites de la connaissance, ou les limites de la raison mathématique. Mais je pense que c'est marrant, donc c'est ce que j'ai fait pendant toute ma vie.

Ainsi, je ne sais pas si c'est pertinent par rapport à SETI ? Je suppose que ça l'est, car si quelque chose a l'air aléatoire, il s'en suit que ce n'est probablement pas issu d'une source intelligente. Mais qu'en est-il si ces êtres supérieurs ont retiré toute redondance de leurs messages ? Il suffirait qu'ils passent tout ça dans un algorithme de compression, car ils nous envoient des messages énormes, ils nous envoient toute leur connaissance, sagesse et philosophie, tout ce qu'ils savent en philosophie, car leur étoile est sur le point de passer en nova, donc c'est un texte énorme retraçant tous les ac-

complissements de leur réflexion et de leur civilisation – et évidemment ils pensent que n'importe quel esprit intelligent traiterait cette information en la compressant, pas vrai ? Et le problème, c'est qu'on se retrouve avec ce message compressé LZ, et on pense que c'est du bruit aléatoire alors qu'en fait, c'est un merveilleux message compact révélant la sagesse et l'héritage de cette grande civilisation ?

GJ/HZ/GO : Oh, mais ils ont inclus l'algorithme de compression ?

Greg Chaitin : Bah, ils peuvent avoir pensé a priori que c'est le seul algorithme de compression concevable, qu'il est si simple que tout être intelligent se servirait de lui pour compresser – je ne sais pas ...

Stephen Wolfram : Je pense que c'est une question intéressante – sur SETI. Par exemple, si on imagine qu'il y a une civilisation suffisamment avancée pour déplacer les étoiles, il y a un type de question intéressante : dans quelle configuration les étoiles auraient été bougées et comment pourrions-nous savoir qu'il y a évidence d'intelligence ayant déplacé ces étoiles ? On a une belle citation philosophique de Kant qui disait "si vous voyez un bel hexagone dessiné dans le sable, vous savez qu'il doit venir d'un certain type d'entité intelligente [qui l'a créé]". Et je pense qu'il est particulièrement charmant que maintenant, depuis quelques années, il est devenu clair qu'il existe des endroits dans le monde où on trouve des arrangements hexagonaux de rochers qui se sont formés, et on sait aujourd'hui qu'il s'agit d'un processus physique qui est la cause de la formation de ces arrangements hexagonaux de pierres. C'est une sorte de version charmante de cela ...

Ed Fredkin : Un des pôles de Saturne possède un bel hexagone – au pôle même, et on a des images de cela.

Stephen Wolfram (continuant) : ... bon, donc la question est que faut-il voir pour croire qu'il y a évidence d'intention, que ça a été fait exprès. Ça me fait penser à Gauss, par exemple, il avait fait le projet de sculpter dans la forêt sibérienne le dessin illustrant le théorème de Pythagore, car on aurait eu alors une chose révélant l'intelligence. Et si on observe la Terre aujourd'hui, une bonne question à poser à un astronaute est : "Que voyez-vous sur la Terre qui vous fasse savoir qu'il y a un genre de civilisation ? Je connais la réponse : ce qui est le plus évident pour un astronaute, c'est – deux choses, d'accord ? La première est : dans le grand lac salé de l'Utah, [il y a] une chaussée qui sépare une région possédant un type d'algue ayant tendance à être orangée, d'une autre région possédant un autre type d'algue qui tend vers le bleuté, et c'est une ligne droite qui sépare ces deux étendues d'eau. C'est parfaitement rectiligne et c'est la chose numéro un. La chose numéro deux se situe en Nouvelle-Zélande. Il y a, en Nouvelle-Zélande, un cercle parfait visible de

l'espace. Je travaillais sur le livre de NKS et j'étais en train d'écrire une note au sujet de cette chose particulière. Nous les avons contacté, c'était avant que le web soit développé comme il l'est aujourd'hui, donc on contacte le New Zealand Geological Survey pour obtenir quelque information sur ce cercle parfait, et ils ont dit : "Si vous écrivez un livre de géologie (le cercle parfait entoure un volcan), s'il vous plaît n'écrivez pas que ce volcan produit ce cercle parfait, car c'est faux." La vérité est qu'il y a un parc naturel qui entoure le volcan, et il se trouve qu'il est parfaitement circulaire, et qu'il y a des moutons qui pâturent dans le parc, mais pas au-delà, donc c'est un cercle produit par l'humain. Mais il est intéressant de voir ce qu'il y a sur Terre qui révèle, en quelque sorte, l'intelligence de sa source.

Concrètement, juste pour faire un commentaire de plus sur l'aléatoire, et sur les entiers – juste pour aborder cette idée globale de savoir s'il existe des entiers aléatoires ou non, si c'est important, comment en être sûr, et le reste – on a un petit projet appelé 'base d'entiers', qui est d'abord un répertoire d'entiers. La question est de trouver le plus simple programme qui produise chacun de ces entiers. Et c'est intéressant, c'est vraiment un projet très pragmatique, essayant de renseigner des programmes concrets fabriquant des entiers. On a du mettre au point une sorte de métrique pour rendre compte de ce qui est simple. Quand on utilise différents types de fonctions mathématiques, on se sert de différents types de constructions de programmes, on doit concrètement décider d'une mesure pour quantifier la simplicité. Et il y a tout un tas de façons de faire, par exemple : combien de fois cette fonction semble être référencée sur le web, ce peut être un critère sur le poids à attribuer ; ou quelle est la longueur du nom de cette fonction ; ou d'autres genre de critères comme ça. C'est donc une sorte de version très concrète de cette question sur les entiers aléatoires.

Tom Toffoli : Je voudrais dire quelque chose qui va donner lieu à un autre corollaire. Il y a ce gourou auto-proclamé de l'électronique, Don Lancaster, il est bien connu dans le milieu, et il a dit quelque chose de très vrai. Il a dit : la pire chose qui puisse arriver à l'humanité, c'est de trouver une source d'énergie inépuisable et gratuite. Vous le savez, on va tous en espérant trouver quelque chose comme ça, mais si on le trouvait, ce serait un désastre, car alors la Terre serait réduite en cendres en très peu de temps.

Si on n'est soumis à aucune des contraintes courantes, ça peut devenir très dangereux. Par exemple, vous avez une maison, et vous avez une montagne, puis vous aimeriez bien avoir du soleil l'après-midi. Et le matin, vous voudriez vous prémunir du froid ou n'importe quoi, donc si l'énergie est gratuite, vous retirez la montagne de là où elle est pendant la matinée, et vous la replantez dans la soirée. Et c'est ce qu'on fait essentiellement quand on fait la navette

entre banlieue et centre de Boston. Vous voyez ce fleuve de véhicules qui se précipite dans un sens chaque jour, et dans l'autre sens le même jour, et toute l'énergie que ça coûte. Imaginez si c'était complètement gratuit. Ainsi, de nouveau, essayez de réfléchir à ce que serait la réponse à notre question si on la trouve pour de bon, et voyez d'abord les conséquences.

Encore une fois : ce commentaire est en rapport avec ce que j'ai dit plus tôt sur le hasard des étoiles, si c'était une indication d'une super-intelligence, ou d'une super-stupidité. Qui sait, il se pourrait que ce soit la même chose ?

Rob de Ruyter : Juste pour prendre notre question tout-à-fait littéralement : il y a beaucoup d'aléatoire devant nos yeux, quand on regarde le monde. Particulièrement dehors, au clair de lune, il y a des photons qui se baladent, mais ils ne sont pas nombreux, et on est très conscient du fait que l'information recueillie par notre système visuel, information à traiter pour naviguer correctement, est de qualité faible, et chose intéressante, c'est que nous, en tant qu'organismes, avons l'habitude de nous déplacer dans un monde aléatoire et nous sommes très conscients de cela. Hier, je parlais de la façon dont les mouches se débrouillent avec ça – on s'en arrange aussi, et à tous les niveaux, de l'adaptation dans les photorécepteurs de nos yeux jusqu'à l'adaptation des algorithmes computationnels que notre cerveau utilise, où que vous soyez dans un environnement où on est sujet à des larges quantités de bruit, car il n'y a pas beaucoup de photons là autour. Alors on a tendance à bouger avec précaution – on ne se met pas à courir, à moins que le tigre soit juste derrière, ce qui est très rare – je pense donc, qu'à tous les niveaux, on a l'habitude que les mesures prises par nos capteurs soient plus ou moins aléatoires, et dépendantes de la situation du moment. Et en tant qu'engins computationnels, on est très conscients de ça.

Cris Calude : Je m'intéresse à la qualité de l'aléatoire quantique. Nous avons été capable de prouver, selon certaines suppositions modérées du modèle quantique de la physique qu'on a accepté, que l'aléatoire quantique n'était pas calculable. Cela signifie qu'aucune Machine de Turing ne peut reproduire le résultat constitué par une séquence de bits quantiques (longue et finie) et cela nous donne une forme affaiblie de la relation entre la théorie de Greg, la définition de Greg de l'aléatoire algorithmique, et ce qu'on considérerait être la meilleure source possible d'aléatoire dans cet univers, c-à-d l'aléatoire quantique. Et une des choses difficiles quand on réfléchit et expérimente, c'est la façon de distinguer l'aléatoire quantique de l'aléatoire généré. Est-il possible, en se servant de façon finie de tests bien choisis, de trouver un marqueur de cette distinction entre quelque chose qui soit généré calculatoirement par un computer, et quelque chose non généré de cette manière ?

Alors qu'ici on a une certaine information sur la source, comme Tom le disait – on sait très bien qu'il existe une définition asymptotique de la façon dont ces bits peuvent être générés – il est toujours très difficile de compter sur une quantité finie de tests pour faire cette différence.

Ed Fredkin : Juste une histoire amusante sur les nombres aléatoires : du temps des premiers ordinateurs, les gens voulaient avoir des nombres aléatoires pour faire des simulations de Monte Carlo et autres choses du genre, donc un beau gros ordinateur fut conçu au laboratoire Lincoln du MIT. C'était le plus gros et le plus rapide ordinateur du monde, il s'appelait TX2 et il devait avoir tous les accessoires, gadgets et options possibles : un écran de rêve et tout le tralala. Et ils ont décidé de se mettre à résoudre le problème des nombres aléatoires, donc ils incorporèrent un registre qui donnait toujours un nombre aléatoire ; cela fut réalisé vraiment soigneusement avec du matériel radioactif, des compteurs Geiger et ainsi de suite. Et donc à chaque fois qu'on regardait ce registre, on obtenait un véritable nombre aléatoire, et ils ont pensé : "C'est une grande avancée pour les nombres aléatoires par ordinateur !" Mais l'expérience donna des résultats opposés à leurs attentes ! L'affaire tourna au désastre complet et tout le monde finit par haïr le dispositif : personne n'écrivit de programmes pouvant le déboguer, car ça ne fonctionnait jamais deux fois de la même manière, ainsi ... Il y avait eu un peu d'exagération, mais au final, tout le monde décida que les générateurs de nombres aléatoires classiques, c.-à-d. le type générateur de séquence par décalage des registres et autres, étaient bien meilleurs. Ainsi cette idée fut abandonnée, et je ne pense pas qu'elle fût jamais reprise.

Stephen Wolfram : Elle est réapparue concrètement, dans la génération courante des puces Pentium, il y a un générateur d'aléatoire hardware basé sur le double bruit de Johnson dans une résistance. Mais en ce temps-là, on pouvait lancer des programmes isolément. Aujourd'hui cela pose problème au sens où on ne peut plus faire tourner des programmes tout seuls : ils ont accès au web, ils font toutes sortes de choses, surtout produire du bruit aléatoire capté à l'extérieur, pas à partir de mécanique quantique mais du bruit aléatoire produit par le monde extérieur. Donc le même problème est de retour.

GJ/HZ/GO : Ca me rappelle des temps obscurs, quand on a publié cet énorme volume – j'en ai trouvé une réimpression intitulée "Cent mille nombres aléatoires" (Cris Calude intervient pour corriger : "Un million de nombres aléatoires") au cas où vous auriez eu un besoin urgent de quelques nombres aléatoires – et Murray Gell-Mann avait l'habitude de raconter l'histoire que lorsqu'il travaillait sur RAND, à un moment, je pense que c'était quand ils travaillaient sur le livre, ils ont dû imprimer une page d'erreurs !

Stephen Wolfram : Cette histoire sur la feuille d'erreurs, je pense que c'est intéressant car ces nombres ont été générés à partir de quelque appareillage de tubes à vide (je pense à une triode, ou autre) et le problème a été quand ils ont commencé à générer les nombres, ils ont essayé de le faire trop rapidement, et à la base, ils n'ont pas attendu que le bric-à-brac restant dans la triode soit dissipé avant de faire la mesure du bit suivant. C'est exactement la même cause de difficulté quand on essaye d'obtenir de l'aléatoire parfait à partir de désintégration radioactive ! Je pense à l'expérimentation laborieuse en cours sur le calcul quantique, dont il peut être intéressant de dire un mot ici, où il y a cette question de comment obtenir – je veux dire, est-il possible d'obtenir – une séquence parfaite de bits aléatoires à partir d'un appareillage quantique ? Qu'est-ce que ça implique de faire ça ? Voilà ce que je suspecte : chaque fois qu'on obtient un bit, on doit passer du niveau quantique au niveau classique de la mesure, on doit, en quelque sorte, propager l'information au sujet de ce bit hors de ce bocal rempli de ce brouillon thermodynamique de matière pour en extraire une mesure précise ; et le litige ou ce que j'en pense, ce serait qu'il y a une vitesse à laquelle cette propagation peut se produire et que finalement, on va avoir des bits qui ne sont pas plus aléatoires que l'aléatoire qu'on pourrait obtenir grâce au seul processus de propagation sans aucune sorte de mini-source quantique. Bon, c'est un point de vue extrême, que le petit morceau (bit) supplémentaire de quantité n'apporte pas vraiment quelque chose à la capacité d'obtention de bits aléatoires. Je ne sais pas si j'ai raison mais vous savez, même si c'était il y a longtemps, j'ai essayé d'examiner certaines expérimentations, et la chose typique que j'ai trouvée, c'est qu'on essaye d'obtenir rapidement des bits aléatoires à partir d'un système quantique et on s'aperçoit qu'on a des fluctuations de bruit $1/f$ à cause des corrélations dans le détecteur et ainsi de suite. Donc je pense qu'au minimum, la question à se poser sur la mécanique quantique est : peut-on véritablement obtenir un genre de bits aléatoires et qu'est-ce qui est mobilisé quand on fait ça ? Que se passe-t-il vraiment dans les appareillages pour faire en sorte que ça se produise ? Je serais curieux de connaître la réponse.

Tom Toffoli : Je connais la réponse, Intel l'a déjà donnée : ainsi, vous pouvez avoir de bons nombres quantiques si vous disposez d'un générateur quantique de nombres aléatoires. Suffit de générer un nombre quantique, puis de jeter votre générateur et d'en acheter un autre, car celui que vous avez utilisé est déjà intriqué avec le nombre qu'il a généré. Donc achetez-en un autre et vous avez la solution du problème. [Wolfram dit : c'est aussi une bonne stratégie commerciale ... les gens rient]

Ed Fredkin : Il y a une bonne histoire dans l'historique de – dans les

années 50, les gens faisant différentes choses électroniques avaient besoin de bruit. Ils voulaient du bruit, du bruit aléatoire, donc ils ont pensé : qu'est-ce qui serait une bonne source pour ça. On s'est alors aperçu qu'un modèle particulier de photomultiplicateur, si on le recouvrait en ne laissant aucune lumière passer, donnait un beau bruit aléatoire. Par conséquent, toute une variété de gens ont mené des expériences, ils caractérisaient ce tube et c'était devenu une source parfaite de bruit aléatoire, et le volume des ventes a commencé à décoller. Différentes personnes ont commencé à construire ces circuits et à les utiliser un peu partout. Pendant ce temps, à la fabrique de tubes qui s'appelait RCA, quelqu'un a remarqué : "Hé, ce vieux photomultiplicateur qu'on avait du mal à vendre avant, les ventes décollent, on ferait mieux de l'améliorer pour qu'il ne soit plus si bruyant !" Ainsi, ils l'ont réparé et ce fut la fin de cette source de bruit aléatoire.

vi information vs matter

GJ/HZ/GO : Je pense que je vais poser une autre question sur une autre chose qui m'a tracassé, avant de commencer à laisser l'assistance réagir. Je suis tombé dessus pour la première fois quand j'écrivais un livre intitulé "Fire of the Mind" dont le sous-titre était "Science, destinée et recherche d'ordre". J'étais en train de relire un livre lu auparavant au collège et qui m'avait vraiment impressionné à l'époque, en voyant qu'il était toujours d'actualité, et qui était le livre de Robert Pirsig "Traité du Zen et de l'entretien des motocyclettes" – il est toujours d'actualité selon moi.

Il y a une scène au début du livre où le protagoniste qui se fait appeler Phèdre, d'après le dialogue de Platon, entreprend un voyage à moto à travers le pays avec son fils Chris – qui, dans la réalité, fut plus tard tragiquement assassiné à San Francisco alors qu'il participait à un monastère Zen, mais cela n'a rien à voir – en tout cas, dans le livre, cette personne voyage avec Chris et une nuit, alors qu'ils sont assis autour d'un feu de camp à boire du whisky, à parler, à se raconter des histoires de fantômes, Chris demande à son père : "Tu crois aux fantômes ?" Et il répondit "Non, bien sûr, je ne crois pas aux fantômes car les fantômes ne contiennent ni matière ni énergie, et selon les lois de la physique, ils ne peuvent exister". Puis il réfléchit un moment et dit : "Bon, bien sûr, les lois de la physique elles aussi ne sont pas constituées de matière ou d'énergie, et donc elles ne peuvent pas exister non plus." Cela me paraît être une très bonne idée, et quand je me suis mis à apprendre la physique computationnelle, ceci m'a fait me demander : où sont les lois de la physique ? Doit-on être platoniste et penser que les

lois de la physique sont écrites dans un certain royaume théorique ? Ou bien cette physique computationnelle nous donne-t-elle les moyens de les concevoir comme incorporées à l'intérieur des systèmes mêmes qu'elles expliquent ? [attend un moment, voit Chaitin qui veut répondre, et dit] Greg !

Greg Chaitin : Georges, on a des fantômes : l'information ! L'information est non-matérielle.

GJ/HZ/GO : L'information est physique, non ? Landauer n'aurait pas dit ça ?

Greg Chaitin : D'accord, peut-être que Rolf a dit cela, mais ontologiquement, on en est à ce nouveau concept d'information, et ceux d'entre nous qui font de la physique digitale mettent en quelque sorte l'information avant la matière. C'est un très vieux débat philosophique : le monde est-il fait d'esprit ou de pensées, ou bien est-il fait de matière ? Qui est le premier, qui est le second ? La vision traditionnelle de ce qu'on voit en tant que réalité, est que tout est fait de matière. Mais un autre point de vue est que l'univers est une idée, et donc (l'information est beaucoup plus proche de cela) fait d'esprit, la matière étant un phénomène second. Donc, prosaïquement, peut-être que tout est fantomatique. Si vous croyez en un modèle computationnel, modèle informationnel de l'univers, alors il n'y a pas de matière ! C'est juste de l'information – des structures d'information dont serait issue la matière.

GJ/HZ/GO : Est-ce que ça vous semble correct, Tony ? Est-ce que ça vous semble juste que l'information soit plus fondamentale que la matière ou l'énergie ? Je pense que la plupart d'entre nous – je veux dire, l'homme de la rue – questionné sur l'information, penserait à l'information en tant que construction humaine imposée à de la matière ou à de l'énergie qu'on fabrique. Mais j'aime vraiment l'idée que Gregory a suggéré – et concrètement, j'ai écrit pas mal de choses là-dessus dans ce livre – que l'information est réellement à la base, et que matière et énergie sont, d'une certaine façon, extraites de l'information.

Tom Toffoli : Bon, dans l'idéal, je veux dire ... Vous pouvez vous poser la même question sur la corrélation plutôt que l'information, car ça véhicule la même signification. De plus, on peut mentionner et discuter de la notion d'intrication, dans le même esprit : ce n'est ni ici, ni là. Ou est-ce ? Problèmes très liés. C'est la remarque que je voulais faire.

Tony Leggett : Eh bien je pencherais plutôt vers l'opinion un peu à courte vue, que l'information me semble être vide de sens – à moins que ce soit une information sur quelque chose. On doit alors se poser la **Question** : "Qu'est-ce que c'est que cette chose ?" J'aimerais penser que ce quelque chose

a quelque chose à voir avec notre affaire, et avec la matière et l'énergie que ça implique.

Stephen Wolfram : Cette question de savoir si les systèmes formels abstraits sont sur quelque chose ou non est une question qui, d'évidence, est venue des mathématiques. Voilà ce que je pense de la réponse à cette question : l'information est-elle la chose primaire, ou c'est la matière qui vient en premier ? Je pense que la réponse à cette question finirait probablement par être que ce sont concrètement les mêmes types de choses. Qu'il n'y a pas de différence entre elles. Que la matière est simplement notre façon de nous représenter les choses, qui sont en fait des structures d'information, mais on peut aussi dire que la matière est la chose primaire et que l'information est juste notre représentation de cela. Ça fait peu de différence, je ne pense pas qu'il y ait une grande distinction à faire – si on a raison de dire qu'il existe un modèle ultime pour la représentation de l'univers en termes de computation.

Mais je pense qu'on peut poser cette question sur le fait de savoir si les systèmes formels sont supportés par quelque chose – cela arrive souvent en mathématiques, on peut inventer un certain système axiomatique et puis déclarer : ce système décrit-il vraiment quelque chose, ou est-il simplement un système axiomatique nous permettant de faire différentes déductions sans concerner vraiment quelque chose. Par exemple, l'une des plus importantes conséquences du théorème de Gödel est que, bien qu'on ait pu penser que les axiomes de Peano ne concernaient que les entiers et l'arithmétique, le théorème de Gödel a montré que ces axiomes admettent aussi différentes variétés d'arithmétiques non-standard, des choses qui ne ressemblent pas vraiment à des entiers ordinaires, mais qui sont quand même cohérentes avec ces axiomes. Je pense qu'il existe réellement une confusion sur la façon dont les mathématiques ont été construites sous leurs formes axiomatiques, ce qui correspond à ce problème d'être "sur quelque chose" pour ainsi dire – et peut-être que tout vient de là. Mais quand on réfléchit à la façon dont on met en place une computation, on a affaire à une règle particulière ; et on dit juste "OK, lançons la règle, et arrivera ce qui arrivera". Les mathématiques ne fonctionnent pas de cette façon, en ces termes, habituellement. On y dit au contraire : voyons voir comment les choses peuvent se passer avec ce système axiomatique comme contrainte. C'est autre chose que de dire, jetons quelques règles sur la table, laissons-les agir, et voyons ensuite les choses qui se passent. En mathématiques, on dit : voilà des axiomes qui doivent à peu près décrire comment les choses fonctionnent, le même raisonnement a été tenu en physique avec les équations – vous savez, l'idée de fabriquer une équation qui décrit ce qui peut arriver – et non pas (comme nous le faisons dans les computations) fabriquons quelque chose où on installera une règle, pour ensuite laisser cette

règle tourner. Ainsi par exemple, c'est la raison pour laquelle en théorie de la gravitation, il y a toute une polémique sur "A quoi ressemblent les solutions aux équations d'Einstein ?" et "Quel est l'ensemble de solutions possibles ?" et non pas (au contraire) "Comment les choses vont fonctionner ?" mais la question traditionnellement posée, c'est "Quelles sont les choses possibles qui soient cohérentes face à ces contraintes ?"

Au final, en mathématiques, on se retrouve avec ces systèmes axiomatiques, et ils essayent d'arranger les choses pour être sûrs qu'ils sont vraiment en train de parler de la chose dont ils avaient imaginé parler au départ, comme les entiers. Et ce qu'on sait d'après le théorème de Gödel et autres, c'est que ce type d'arrangement ne peut jamais réellement marcher. On ne peut jamais vraiment utiliser ce modèle basé-sur-la-contrainte sur comment comprendre les choses, pour réellement faire en sorte que notre compréhension se porte sur une certaine chose. Je pense donc que c'est une sorte de problème ultime que cette idée de savoir si les lois qu'on utilise pour décrire les choses et les choses elles-mêmes sont une seule et unique entité ou si elles sont différentes, et si c'est le cas, quelle est vraiment cette distinction.

Greg Chaitin : Il y a cette vieille idée que peut-être le monde est constitué de mathématiques, et que la réalité ultime est mathématique. Par exemple, les gens y ont pensé avec les mathématiques continues, avec les équations différentielles et les équations différentielles partielles, et ce point de vue a eu énormément de succès, c'est donc déjà une vision non-matérialiste du monde. Je voudrais dire aussi que la mécanique quantique n'est pas une théorie matérialiste ; quelle que soit l'équation d'onde de Schrödinger, ce n'est pas de la matière, donc le matérialisme est définitivement mort, autant que je sache. La manière dont Bertrand Russell l'a amené est : si vous prenez le point de vue disant que la réalité n'est que ce que sont les apparences de tous les jours, alors que la science moderne montre que la réalité quotidienne n'est pas la réalité réelle, donc – je ne sais plus comment il appelait ça ... 'réalisme naïf', je pense – si le réalisme naïf est vrai alors c'est faux, donc il est faux. C'est une autre façon de la représenter. Ainsi, la seule chose que nous avons changée avec cette vision disant que la structure réelle du monde est mathématique, la seule chose nouvelle qu'on a ajoutée aujourd'hui, c'est que nous décrivons les mathématiques comme finalement computationnelles, ou que finalement ça concerne l'information, les zéros et les uns. C'est un léger raffinement d'un point de vue assez classique.

Donc ce que je dis, c'est que d'une certaine façon, nous ne sommes pas aussi révolutionnaires qu'il n'y paraît, c'est juste l'évolution naturelle d'une idée. Autrement dit, cette question de l'idéalisme versus le matérialisme, ou "le monde est-il constitué d'idées, ou le monde est-il constitué de matière ?",

ça pourrait avoir l'air fou, mais en fait c'est la question "la structure ultime du monde est-elle mathématique?" versus la matière. Ça a l'air moins théologique et plus les pieds sur terre. Et on a une nouvelle version de cela, vu que les idées continuent à se mettre à jour, la forme courante de cette nouvelle version est simplement : "est-ce de la matière ou est-ce de l'information?" Ainsi, ce sont des idées anciennes qui changent de forme avec le temps, elles évoluent, elles se recyclent, mais elles sont toujours visiblement pas si éloignées de leur origine.

Cris Calude : Information versus matière, discret versus continu : intéressants contrastes philosophiques, et j'ai trouvé la description de Stephen extrêmement intéressante, car je pense aussi que ces points de vue doivent en fait coexister dans une dualité productive. Et cela dépend de nos propres capacités, cela dépend des problèmes, si l'une des options devient plus visible ou plus pratique. En fin de compte, ce qui est vraiment important est – si vous avez un point de vue qui fait prévaloir l'information dans un problème spécifique – qu'est-ce que vous obtenez à partir de cela ? Pouvez-vous prouver une théorie, pouvez-vous obtenir un certain résultat, pouvez-vous construire un modèle sachant qu'il va répondre à une question importante, ou non ? Dans certains cas, un point de vue peut être le bon, dans d'autres cas ce sera l'autre. Donc d'après moi, je dirais de façon plus pragmatique : regardez, choisissez le point de vue que vous voulez de façon à obtenir un résultat, et si ça marche, cela veut dire que pour ce problème-là, ce choix a été le bon.

Ed Fredkin : Mais en fait, le monde est soit continu, soit discret, et on peut appeler ça comme on veut, obtenir des résultats et autres, faire des arrangements – mais il y a réellement une réponse à cela, et l'une des réponses est vraie et l'autre est fausse. Je vais dans le sens de Kronecker qui a dit "Dieu inventa les entiers et tout le reste est œuvre humaine". Ainsi, vous pouvez faire n'importe quoi avec des modèles discrets et/ou des modèles continus, mais cela ne signifie pas que le monde est les deux ensemble, ou peut l'être, ou pourrait l'être. Non, le monde est soit l'un, soit l'autre.

Stephen Wolfram : Toutes ces questions sur le mécanisme sont assez amusantes. Par exemple avec les modèles basés sur les ACs, les gens de la physique traditionnelle (c'est différent dans les autres domaines, seulement les gens de la physique traditionnelle) ont souvent considéré ces types de modèles avec beaucoup de scepticisme. C'est une espèce de retournement historique amusant car, pendant la période pré-newtonienne, les gens ont toujours eu des modèles mécanistes pour décrire les choses – que ce soit des anges poussant la Terre sur son orbite, ou autres genres de choses mécaniques. C'est alors qu'est arrivé ce type de mathématiques purement abstraites décrivant la loi de gravité et autres, et tout le monde a dit – mais seulement après un moment !

– “Bien, tout ça n’est que mathématique et il n’y a pas de réalité matérielle, il n’y a pas de mécanique derrière ces choses!” Et ainsi, c’est quand on débarqua avec ces modèles computationnels qui ressemblent beaucoup plus à de la mécanique tangible, que ça a paru louche et presque non-scientifique aux gens qui avaient passé leur vie à travailler à l’intérieur du paradigme mathématique, que c’était trop simple d’avoir un mécanisme compréhensible derrière les choses. Ainsi on a affaire à une sorte de curieux renversement au cours de ce processus historique. Comme vous le savez, j’ai travaillé sur un tas de choses différentes, et comme je l’ai dit, je n’ai pas travaillé beaucoup à chercher une théorie fondamentale de la physique, ces derniers temps. Je trouve vraiment que ce débat excite ma motivation et j’ai bien envie de me lancer concrètement à la recherche d’une telle théorie fondamentale de la physique. Parce que je pense que, d’une certaine façon, avec toutes ces espèces de questions métaphysiques sur ce qui devrait être là, et ce qui ne devrait pas l’être, et ainsi de suite : damned, on peut en fait répondre à ces questions.

Tom Toffoli : Je connais un moyen de s’en sortir ! Cela ressemble à ce que Ed a proposé il y a longtemps. Il a dit (au sujet de la calculabilité, en faisant référence aux problèmes exponentiels, polynomiaux) qu’on pouvait transformer tous les problèmes en problèmes linéaires. Vous voyez, tous les problèmes exponentiels ! Et bien sûr, les gens ont dit : “Ed, vous êtes un amateur indiscipliné, vous dites ça sans savoir de quoi vous parlez.” Mais il a répondu : “Regardez, on a la loi de Moore ! Et d’après la loi de Moore, tout double de vitesse tous les tant d’années. Donc il suffit d’attendre suffisamment longtemps et on aura la solution – aussi longtemps que ça prendra.”

Ainsi la solution-clé serait d’attendre aussi longtemps qu’il faudra. Avec ça, je vais maintenant donner une solution plus simple à ce problème, sur trouver une théorie fondamentale de la physique, en partant de l’observation que les animaux domestiques deviennent moins intelligents que les animaux sauvages. Cela a été prouvé récemment lors de recherches sur les loups. Peut-être que les animaux domestiques sont, d’une certaine façon, moins intelligents mais ils peuvent voir ce que les humains désirent et ils obéissent. Mais ensuite on a fait certaines expériences supplémentaires et les résultats furent que les loups sauvages, une fois placés dans un environnement avec des humains, apprennent les humains plus vite que les animaux domestiques, pour ce qui est d’anticiper la volonté de leur entraîneur. Cela ne veut pas dire qu’ils suivent ses volontés, mais ils les anticipent plus vite.

Aujourd’hui, on est en train de faire une grande expérience sur nous-mêmes, sur l’humanité – l’humanité est en passe de se domestiquer elle-même. Il fut un temps où les gens disaient : nous avons découvert les équations

différentielles et très peu de gens peuvent le comprendre, donc nous avons le monopole, nous sommes des scientifiques. Et finalement quelqu'un a écrit "Bon, attendez une minute, mais pourquoi ne pouvons-nous pas trouver un modèle comme celui de Ed ou de Stephen – qui soit une sorte de modèle computationnel, discret, que tout le monde puisse se l'approprier et posséder etc ?" Mais on oublie qu'on est en train de se domestiquer nous-mêmes ! Au point que l'ordinateur – selon Brian Hayes, la première chose concernant l'ordinateur, c'est qu'on peut le programmer et lui faire faire ce qu'on veut – au point qu'aujourd'hui la plupart des gens ne savent même pas que c'est possible, ils conçoivent juste l'ordinateur comme un appareillage. Bientôt l'ordinateur deviendra un mystère pour la plupart des gens ! L'ordinateur digital programmable va subir le même sort que les équations différentielles il y a une génération – ainsi le problème est résolu, en un sens, il suffit juste d'attendre assez longtemps et plus personne ne sera capable de s'occuper de ces choses, ce sera juste un mystère pour nous.

Cris Calude : Bon, je voudrais seulement faire une petite remarque, inspirée par votre idée sur la computation physique. Donc, c'est surtout une remarque personnelle sur le problème P versus NP : je crois que c'est une question mathématique très stimulante, et profonde et intéressante, mais je pense qu'elle n'a pas la moindre signification informatique. A cause du simple fait que P n'est pas un modèle adéquat de computation réalisable, et il existe de nombreux résultats – à la fois théoriques et pratiques – faisant ressortir que P ne modélise pas convenablement ce que nous entendons par computation réalisable. Probablement le plus simple exemple est de réfléchir à l'algorithme simplex, qui est exponentiellement difficile, mais qui fonctionne bien mieux en pratique que toutes les solutions polynomiales connues.

vii Questions du public, sans modérateurs

Georges Johnson : C'est le bon moment pour passer aux questions du public.

Jason Cawley : J'ai une question pour Greg Chaitin, mais pour tous les autres aussi. Vous avez dit que le monde serait plus intelligible et plus joli s'il était discret, ce qui m'a bien plu. Mais jusqu'où serait-il intelligible, concrètement, même si je vous accorde finitude et discrétion, même si nous trouvons les règles ? N'y aurait-il pas toutes ces poches de complexité en lui, ne seraient-elles pas énormes comparées à nous, finis, ne dureraient-elles pas beaucoup plus longtemps que nous – et on aurait alors encore tous ces

types de cheminements qui nous seraient mystérieux dans tous leurs moindres détails ?

Greg Chaitin : Oh, je pense que cette remarque est en désaccord avec ce que j'ai dit sur un univers discret qui serait plus beau ... non, plus compréhensible ... c'est ça ? ... et vous avez donné tout un tas de raisons pour lesquelles vous pensez qu'il serait affreux, incompréhensible, dégoûtant – et je n'y peux rien si vous le sentez comme ça ! Dans ce cas, je ne pense pas que ce soit une question, je vois ça comme un commentaire qui n'attend pas de réponse ...

Jason Cawley : Pardon. La question est “Jusqu'où un univers parfaitement discret est-il intelligible ?” La raison pour laquelle je m'intéresse à cela est : j'aime trop le rationalisme pour laisser les gens conclure à son absence lorsqu'ils voient de la non-intelligibilité dans l'univers.

Greg Chaitin : D'accord ! Bon, dans ce cas, comme Stephen l'a signalé dans son livre, il se peut que tout l'aléatoire dans le monde soit juste du pseudo-aléatoire, voyez-vous, et que les choses paraissent seulement intelligibles, tout en étant réellement rationnelles. L'autre chose qu'il a aussi signalée – et c'est en quelque sorte sa version du théorème d'incomplétude de Gödel – c'est que quelque chose peut être simple et discret, et pourtant on sera incapable de rien prouver sur ça. Et la version de Stephen à ce sujet (qui est très intéressante, je pense) est que, de la façon dont l'univers est créé, et parce qu'une computation est quelque chose qui se lance, on doit généralement lancer un système physique pour voir ce qu'il va faire, il n'y a pas de raccourci pour la réponse¹⁸. Donc ça peut se voir comme une mauvaise nouvelle, car ça signifie aussi qu'on peut avoir une théorie simple du monde qui ne va pas aider beaucoup pour faire des prévisions. On peut aussi voir ça comme une bonne nouvelle, car ça signifie que l'évolution temporelle de l'univers est créatrice et capable de surprises, ça fait des choses qu'on ne peut pas connaître à l'avance – juste en s'asseyant devant un bureau pour y réfléchir – je vois donc ça comme fondamental et créatif ! Et par rapport à la créativité, Bergson parlait il y a 100 ans de “L'Evolution Créatrice” – ce pourrait être une nouvelle version de cela. Mais finalement, sur l'esthétique, on ne peut pas trop argumenter. Encore une fois, je pense que c'était une bonne question.

Nouvelle question du public : D'abord une question sur le SETI : je suis peut-être naïf, mais il me semble étrange de devoir observer des cercles parfaits et autres lacs divisés quand on peut simplement observer les émana-

18. D'après le Principe d'Irréductibilité Computationnelle (qui est un corollaire du PCE-Principe d'Equivalence Computationnelle).

tions de lumière terrestre de Chicago et New York pendant la nuit. Si j'étais un martien, c'est ce que je ferais. Mais ce qui m'intéresse plus et c'est une question pour M. Leggett, est ceci : supposons qu'il y ait des transitions de phases du quantique au classique. Construire un ordinateur quantique – les approches pour construire un ordinateur quantique seraient-elle différentes de celles qui consistent à essayer de réduire la décohérence comme on le fait à présent ?

Tony Leggett : Je n'ai pas entièrement compris ce que vous voulez dire par postuler qu'il y ait une transition de phase du quantique au classique. Pouvez-vous détailler un peu ?

Question : Bon, eh bien, je suis un peu ignorant de votre théorie mais s'il existe une théorie qui ne soit pas juste de la décohérence, mais qu'il y ait en fait des phases de transition du quantique au classique à un certain niveau – ce qui serait pourquoi on ne voit plus le chat de Schrödinger après un certain temps. Cela pourrait-il impliquer une approche différente quant à la construction d'un ordinateur quantique ?

Tony Leggett : Si vous voulez dire – si vous faites référence à des théories, par exemple du genre GRWP qui postule l'existence de systèmes mécaniques physiques qui rejoindront le formalisme linéaire de la mécanique quantique, qu'on devra modifier – et il faudra le modifier encore plus lorsqu'on passe du microscopique au macroscopique – alors je pense que la réponse à votre question est que tant qu'on voudra utiliser des systèmes macroscopiques ou demi-macroscopiques comme les qbits dans notre ordinateur quantique, ça ne devrait pas marcher. D'un autre côté, je ne pense pas que les théories du GRWP – scénarios du type GRWP – nécessairement opposés à la tentative de construction d'un ordinateur quantique, prennent toujours les bits individuels à un niveau microscopique. On doit les observer à ce niveau, bien sûr, et en détails dans le contexte spécifique d'un ordinateur particulier, tel que par exemple l'algorithme de Shor. Je ne vois pas le fait qu'un scénario du type GRWP détruirait cette possibilité, comme un a priori essentiel.

Nouvelle question du public : Bien, nous n'avons toujours pas répondu à la question "Comment la Nature compute/calcul ?" On a juste débattu des différences entre continu et discret, mais si on voit les mathématiques comme un accident historique et qu'on essaye de les mettre de côté, ne devrions-nous pas essayer d'observer la nature, en essayant de comprendre comment en fait elle calcule ? Et pas seulement essayer de la traduire dans un contexte mathématique. Par exemple, on peut observer les plantes et voir qu'il y a une espèce de type parallèle de computation qui est à l'œuvre, qui n'est pas basé sur nos mathématiques, mais c'est comme si elles le faisaient, les plantes ...

voyez-vous ... que pensez-vous de cela ?

Stephen Wolfram : Je pense que l'une des choses qui rend cela concret est la question de savoir comment on doit construire des appareillages utiles à la computation. Si nos appareillages computationnels pratiques doivent contenir en eux des ALUs (Arithmetic Logic Unit) ou non. Le seul fait que tous les CPU fabriqués possèdent une partie qui s'appelle "l'unité arithmétique logique", nous dit que dans notre conception courante de la computation, il y a des mathématiques quelque part à l'intérieur. D'où la chose intéressante suivante : pouvons-nous réaliser des tâches computationnelles utiles sans avoir à mettre un ALU dans le coup, pour ainsi dire. Je pense que la réponse est définitivement oui, mais comme toute l'ingénierie du développement des ordinateurs s'est passée comme ça jusqu'à présent, on n'a fait qu'optimiser ce modèle particulier qui est basé sur les mathématiques. Et si on essaye de construire des ordinateurs plus proches de l'échelle moléculaire, on pourrait finir par se servir du même modèle : reprendre le design de la puce Pentium, le rétrécir énormément, puis l'implémenter avec des atomes. Mais il y a une alternative, ce serait de pouvoir avoir des atomes qui fassent des choses qu'ils ont l'habitude de faire, pour lesquelles ils sont plus naturellement disposés. Et je pense que cela devrait apparaître en premier lieu dans des choses comme les drogues algorithmiques, où on veut avoir quelque chose qui soit essentiellement une molécule opérant dans un certain contexte biologique, biomédical, et on veut que ça fasse une computation concrète pour que ça comprenne si ça doit se lier à un certain site ou non, ce qui est opposé au discours "je suis de la bonne forme, donc je vais m'accrocher là !" C'est donc un lieu où de la computation doit se produire. Mais ça ne va pas être de la computation passant par de l'arithmétique, et on est obligé de concevoir dans ses propres termes cette computation à l'échelle moléculaire, simplement parce que c'est à l'échelle à laquelle la chose doit opérer. Et j'imagine qu'il y aura des séries entières de choses et d'appareillages – surtout pilotés par une situation moléculaire – où on voudra faire de la computation mais où cette computation ne voudra pas passer par la couche intermédiaire de l'arithmétique.

Cris Calude : Oui ! Il existe beaucoup de recherches qu'on a commencées il y a une dizaine d'années à Auckland, et une série de conférences intitulées "Computation non-Conventionnelle". C'est une des questions intéressantes. Voilà, à la base, il y a deux courants de pensée : l'un est la computation quantique, et l'autre est la computation moléculaire. Et dans la computation quantique, vous avez cette tendance à utiliser les mathématiques à l'intérieur. Mais dans la computation moléculaire, voyez-vous, ça part complètement dans la nature parce que les mathématiques sont absentes donc on se sert de

toutes sortes d'opérations biologiques spécifiques pour la computation. Et si vous jetez un œil aux résultats, certains d'entre eux sont assez spectaculaires.

Petit raffinement : Ouais, mais c'est toujours basé sur les portes logiques et ... vous savez ... la computation moléculaire est toujours basée sur la tentative de construction de portes logiques et ...

Cris Calude : Non, non ! Ce ne sont pas des portes logiques ! C'est ça la différence, car la philosophie de la computation quantique est : vous faites ces portes logiques au niveau des atomes ou autres particules – mais en computation moléculaire, il n'y a pas d'instruction mathématique, il n'y a pas de nombres, voyez-vous, tout est chaînes, et la façon dont elles sont manipulées se base exactement sur un type biologique de processus. Pas d'arithmétique.

Greg Chaitin : Pas d'algèbre booléenne, pas même des 'et' et des 'ou' ?

Cris Calude : Non, rien ! Et c'est la beauté de la chose, et c'était d'une certaine façon la question que j'ai posée à Seth Lloyd en 98. Je lui ai dit tu sais, pourquoi vous ne faites pas quelque chose de semblable en computation quantique ? Pourquoi vous n'essayez pas de réfléchir à certaines sortes de règles de processus – non-imposées par la computation classique, par les machines de Turing, mais des règles qui viennent naturellement des processus quantiques – tout comme l'approche typique en computation moléculaire.

Tom Toffoli : Je voudrais donner un complément de réponse. J'ai enseigné au sujet des microprocesseurs et les microcontrôleurs – vous les avez dans vos montres et téléphones cellulaires. Ce sont des objets extrêmement compliqués. Et on pourrait dire, étant donné la facilité avec laquelle on peut fabriquer ces choses, qu'à chaque fois qu'on veut lancer un programme ou un algorithme, on pourrait fabriquer un calculateur spécialement dédié plutôt que de se servir d'un microprocesseur pour les programmer. Mais apparemment, il se trouve que c'est beaucoup plus commode, si quelqu'un a conçu un microprocesseur avec un ALU, un cache et d'autres choses encore, de juste les prendre comme déjà là, et l'affaire est réglée. Si on regarde la biologie, la biologie a fait la même chose. A un certain moment, on va avoir des mitochondries détournées qui vont faire la conversion de l'oxygène et du sucre pour recharger les batteries ATP. Ca a été une grande invention. Et maintenant, après probablement trois milliards d'années ou à peu près, on s'en sert toujours au lieu d'avoir inventé une méthode qui serait peut-être légèrement plus efficace mais qu'on devrait choisir très précisément pour chaque circonstance. En gros, on pourrait optimiser plus, mais ça nous coûterait une perte de flexibilité, de modularité et ainsi de suite, et apparemment c'est beaucoup plus commode comme c'est. Pendant trois milliards d'années, on

a gardé ce genre de microprocesseur énergétique, qui a très bien fonctionné ... C'est donc essentiellement la souplesse et la modularité évolutive qui inspirent concrètement des choix comme l'ALU ... ce n'est pas un choix optimal, mais – empiriquement – c'est un très bon choix. C'est mon point de vue.

Stephen Wolfram : Bon, comme c'est toujours le cas dans l'histoire de l'évolution technologique, il est inconcevable de revenir en arrière pour faire une sorte de redémarrage de tout le processus de conception. Simplement parce qu'il y aurait beaucoup trop d'investissement à faire à chaque fois. Le point de vue que je veux souligner est que ce qui va se passer, c'est qu'il va y avoir certains problèmes technologiques particuliers, qui vont conduire vers différents types de computation. Mon idée est que les premiers à prendre vraiment de l'importance seront les problèmes biomédicaux, car ils doivent opérer à une échelle moléculaire, car c'est l'échelle à laquelle la biomédecine opère. Et, vous savez, on peut partir vers des dispositifs beaucoup plus gros pour d'autres objectifs – mais la biomédecine est un endroit où une décision doit être prise par une molécule. Et s'il y aura moyen d'un peu pirater autour de ça, il ne devrait pas falloir attendre longtemps avant que les premières applications soient finalisées.

Et ce qui est intéressant là-dedans, c'est que si vous regardez le temps passé depuis Gödel et Turing jusqu'au moment où les ordinateurs deviennent génériques et que tout le monde en a un, puis le temps passé depuis Crick, Watson et l'ADN jusqu'au moment où la génomique devient générique – c'est à peu près le même intervalle de temps. Ce n'est pas encore arrivé pour la génomique, c'est plus ou moins arrivé pour les ordinateurs. Les ordinateurs furent inventés, l'idée des ordinateurs est venue vingt-trois ans ou à peu près, avant l'idée de l'ADN. En tout cas, ça va arriver bientôt aussi pour la génomique, et on sera bientôt capable de séquencer couramment, en temps réel, de nous-mêmes, et on va faire toutes sortes de prédictions que oui, on sait qu'on a détectées aujourd'hui que vous avez une plus haute population d'anticorps, dotés de telle forme particulière, ainsi on sera capable de lancer une certaine simulation qui signifiera que vous devrez vous procurer un apport de cellules T, qui ont telle caractéristique et ainsi de suite. La question reste de savoir si les décisions à ce sujet vont être prises extérieurement par des ordinateurs basés sur des ALUs, ou si ce sera fait intérieurement par une sorte d'appareillage moléculaire – plus proche de ce que fait concrètement la biologie. Et si ça finit par être fait extérieurement, alors il n'y aura aucune impulsion de la part de la technologie pour fabriquer une sous-structure différente pour la computation.

Nouvelle question du public : J'étudie l'informatique. Et l'une des idées que je trouve vraiment fascinante et je pense qu'elle l'est – c'est ce

qu'on appelle l'isomorphisme de Curry-Howard, qui traite des liens entre propositions et types, entre règles et termes, et entre normalisation de preuves et évaluation de programmes. Et comme le débat porte sur les modèles de computation, je me demandais si vous aviez rencontré quelque chose de semblable avec les automates cellulaires et associés. Je pense que la classification, que cette relation particulière est très explicite dans le lambda-calcul de type simple où les termes des programmes, qui sont des expressions lambda, peuvent être des types donnés – et les types sont essentiellement des expressions propositionnelles logiques. Et si vous pouvez prouver une certaine proposition, la structure de la preuve dans son style naturel de déduction, va vraiment ressembler à un type de programme et si la preuve n'est pas une preuve normale, alors le processus de normalisation de preuve est fondamentalement un processus d'évaluation des termes par rapport à la forme normale des termes, ce qui veut dire en fait que si vous avez comme modèle computationnel le lambda-calcul, les réductions du calcul vont correspondre aux normalisations de preuves, et les types servent de moyen de classification pour les programmes, les types sont un moyen de dire : ces programmes particuliers se comportent de telle ou telle façon, ne possèdent pas telle ou telle propriété. Et j'ai l'impression que cela pourrait aussi s'appliquer à d'autres notions de computation, car il n'y a rien d'intrinsèque au sujet du lambda-calcul qui fasse que cela ne s'applique qu'à lui.

Stephen Wolfram : Laissez-moi commencer par vous dire que je suis quelqu'un de très anti-type. Savez-vous qu'historiquement, les types ont été inventés comme découpage par Russell pour, en fait, éviter certains paradoxes – les types sont alors devenus une sorte de "grande chose" qui fut prise en exemple, puis, en ingénierie pratique des premiers langages informatiques, il y a eu cette idée des types d'entiers opposés aux types réels et ainsi de suite, et l'idée même de types prit beaucoup d'importance – du moins, je le vois comme ça.

Ainsi par exemple, dans Mathematica, il n'y a pas de types. C'est un système symbolique où il n'y a qu'un type : l'expression symbolique. Et en informatique pratique, l'usage le plus convaincant des types seraient les formes variées de vérification, mais d'une certaine façon, quand quelque chose est vérifiable par les types, ça implique une certaine sorte de rigidité dans les programmes qu'on peut écrire, ce qui restreint en quelque sorte l'expressivité du langage que vous avez. Et ce qu'on a trouvé maintes et maintes fois dans Mathematica, quand on a pensé organiser les choses en types, c'est que ça aurait effectivement enlevé la possibilité de toutes sortes de paradigmes intermédiaires de programmation, pour ainsi dire, qui existent quand on n'a pas vraiment de types.

Ayant dit ça, cette question sur l'analogie entre les processus de preuves et les processus de computation est une question intéressante. J'y ai beaucoup réfléchi, et il y a pas mal de choses à dire là-dessus. Mais il y a une chose à laquelle il faut penser : "Qu'est-ce qu'une preuve?" et "A quoi bon fabriquer une preuve?" Je veux dire que le rôle réel d'une preuve est de fournir un moyen de convaincre (des humains, au fond) que quelque chose est vrai. Car lorsqu'on fait une computation, on se contente de suivre certaines étapes de cette computation en supposant que l'ordinateur fonctionne correctement, de façon à ce que le résultat sorte selon des règles particulières données pour cette computation. L'intérêt d'une preuve est, plus ou moins, d'être capable de dire à un humain – regardez cela : vous pouvez voir toutes les étapes et vérifier que c'est correct. Je pense que le rôle des preuves, dans les temps modernes, est devenu au mieux un peu bizarre. Car, par exemple, voici un cas typique : quand Mathematica est né il y a vingt ans, on aurait foncé voir les mathématiciens, qui auraient dit : "Comment puis-je me servir de ça, je ne peux pas prouver que tous les résultats obtenus sont corrects !" OK ? C'est ça qui les intéressait. Je voudrais souligner que lorsque vous pensez détenir une preuve, dans certains journaux par exemple, elle a peut-être été vérifiée par une personne – peut-être – si vous avez de la chance. Dans Mathematica, on peut automatiser la vérification de nombreuses choses, et on peut obtenir une assurance automatique de qualité. C'est une règle générale que – en terme de niveau de confiance à accorder – plus il y a de gens qui utilisent la chose dont vous vous servez, plus il est probable que tous les bugs vont être trouvés au moment où vous allez l'utiliser. Donc si vous dites : "Bah, peut-être qu'il y a un problème dans le logiciel que j'ai écrit moi-même, peut-être que c'est un problème dans le système (comme Mathematica) dont je me sers, ou peut-être un problème dans le hardware de l'ordinateur" – il devient de moins en moins plausible qu'il y ait un problème, plus la chose est utilisée.

Quand quelqu'un dit "Je veux une preuve", ça me fait penser que la demande de preuves, au moins le genre de choses que Mathematica fait, s'est atrophiée dans les quelques premières années de l'existence de Mathematica, car il était devenu clair que les fautes les plus probables arrivent quand un humain essaie d'expliquer à l'ordinateur ce qu'il a à faire. Maintenant ceci dit, il est intéressant de voir que dans Mathematica, il y a de plus en plus de types d'objets qui sont essentiellement des systèmes de démonstration – différentes sortes de choses, par exemple des systèmes de preuves pour l'algèbre réelle, on vient d'ajouter dans le grand nouveau Mathematica 7.0 toutes sortes de matériel pour faire de la computation avec des centaines de milliers de variables et ainsi de suite. Ces matériels sont effectivement l'endroit où on a ajouté des systèmes de preuves pour ce genre de choses. On

a ajouté aussi un système de preuve généraliste pour la logique équationnelle, mais encore une fois, je pense que cette question de savoir si les gens sont intéressés par une preuve, ou s'ils veulent juste des résultats – il semblerait que la demande de présentation de preuve soit très basse.

Tom Toffoli : Si vous me donniez le micro pour une seconde : pour en revenir à Russell quand il publia les *Principia Mathematica* – la plupart des théorèmes étaient justes, mais une bonne portion des preuves se révélèrent fausses. C'était Russell, pas vrai ? C'était faux, mais ce n'était pas un problème, car il était toujours convaincu, il s'était fait une conviction par ses propres moyens sur ces théorèmes. Donc il a mis bout à bout quelques preuves (arguments) pour essayer de convaincre les lecteurs que les théorèmes étaient justes, et il les a convaincus. Mais les preuves, en tant qu'outils mécaniques concrets, ne fonctionnaient pas. Ainsi ses preuves étaient devenues un simple appareillage heuristique d'où il dérivait, et vous pouvez toujours dériver le bon théorème à partir d'une preuve fautive. Que faire dans ce cas ?

Stephen Wolfram : Non, je pense que c'est vraiment intéressant toutes ces analogies entre les preuves et les computations et le reste. Une des choses que j'ai souvent remarquée, c'est que si vous regardez les premières tentatives des gens pour formaliser les mathématiques – les choses importantes à formaliser pour eux étaient les processus de démonstration, et c'est sur quoi ont travaillé Whitehead, Russell, et Peano avant lui, et d'autres. Cette orientation de la formalisation s'est avérée assez aride. Peu de choses en sont sorties. L'orientation de la formalisation qui s'est trouvée être la plus intéressante a été, en fait, la formalisation des processus de computation. Ainsi, dans la construction de Mathematica, on a essayé de formaliser le processus de computation des choses. Beaucoup de gens l'ont utilisé et ont fait des choses intéressantes avec. La proportion des gens qui font de la computation avec des mathématiques formalisées, par rapport au nombre de ceux qui font de la démonstration avec des mathématiques formalisées, est énorme. Le versant preuve s'est avéré ne pas être si intéressant que ça. Même genre de choses, et question intéressante, comment arriver à formaliser le langage de tous les jours : on peut prendre le langage quotidien et en donner une version formalisée exprimant les choses d'une d'espèce de façon formelle, symbolique. Mais la chose que je n'avais pas comprise – aujourd'hui, je pense que je l'ai comprise, mais pas à l'époque – était "A quoi bon faire ça ?" Autrement dit, les efforts de Russell-Whitehead pour formaliser les preuves n'ont pas mené bien loin. La bonne idée quant à la formalisation mathématique, a été de formaliser le processus de computation. De même, formaliser le langage quotidien comme moyen de fabriquer le web sémantique et tant d'autres choses, cache probablement le même genre de problèmes que la formalisation de preuves

quand elle fut faite en mathématiques, et je pense que peut-être, un autre cheminement sera suivi lors de la formalisation du langage de tous les jours, et il y a une intéressante analogie à faire, en comparant avec ce qui s'est passé dans le cas des mathématiques. Vous voyez : à quoi bon formaliser, et que peut-on faire avec un système formalisé comme ça ?

Greg Chaitin : Laissez-moi répéter cette remarque très profonde que Stephen vient juste de faire sur les preuves opposées à la computation : si vous examinez n'importe quel système formel destiné à une théorie mathématique formelle, Gödel a montré en 1931 que celle-ci serait toujours incomplète. Donc tout langage artificiel destiné à faire des mathématiques sera incomplet, ne sera jamais universel, ne possédera jamais tous les arguments mathématiques possibles. Il n'existe pas de langage formel pour les mathématiques, dans lequel on pourrait écrire toute preuve ou tout argument mathématique possible. Savez-vous que la théorie de Zermelo-Fraenkel, ceci dit en corollaire de Gödel – peut bien avoir l'air merveilleuse pour tout ce que nous avons, mais c'est incomplet. Maintenant l'exact opposé – la terminologie est différente – quand vous parlez d'un langage de programmation, vous ne parlez pas d'incomplétude et de complétude, vous parlez d'universalité. Et la chose étonnante, c'est que ce désir de formalisme, et Russel-Whitehead est une donnée de référence là-dessus, une autre donnée de référence est le programme de Hilbert, la quête de la formalisation prenait naissance en mathématique, et l'idée était de formaliser le raisonnement ; et la chose étonnante, c'est que ça a échoué. Gödel en 1931 et Turing en 1936 ont montré l'existence d'obstacles fondamentaux – ça ne peut pas marcher ! Mais la chose étonnante, c'est que c'est un merveilleux échec ! Je veux dire que ce qui peut se formaliser parfaitement, ce n'est pas la preuve ou le raisonnement, mais c'est la computation. Et là presque tous les langages avec lesquels vous vous ramenez sont universels, ce qui revient à dire : complets – car tous les algorithmes peuvent s'exprimer en eux. C'est une façon d'amener ce que disait Stephen.

Donc le rêve de Hilbert, Russel, Whitehead a glorieusement échoué – ce n'est pas bon pour le raisonnement mais bon comme technologie, voici un autre moyen de le présenter, si vous essayez de faire de la provocation. La quête d'un fondement solide pour les mathématiques a échoué, mais elle a donné naissance à une industrie valant des trillions de dollars.

Tom Toffoli : Laissez-moi ajouter quelque chose à cela. Vous avez entendu parler de la loi de Parkinson, celle qui dit : "un système va se servir de toutes les ressources disponibles." Elle fut formulée car on remarqua que tout l'Empire Britannique avait été mené en sous-main par quelques douzaines de personnes à peu près, pendant deux siècles. Puis quand les Britanniques ont commencé à perdre leur Empire, ils commencèrent à décoloniser etc., puis ils

eurent un ministre des colonies et ce ministère devint de plus en plus gros à mesure que les colonies devenaient de plus en plus petites. Et ça ne marchait pas aussi bien qu'avant. Je pense que la formalisation se passe souvent comme ça. Vous pouvez y penser, mais vous ne désirez pas réellement la faire, même von Neumann, voyez-vous, le jour où il décida que les ACs étaient, en quelque sorte, des candidats plausibles pour donner la vie – il ne s'attela pas au processus de développement global de la chose. L'essentiel était déjà fait.

Nouvelle question du public : Si c'est possible, puis-je ajouter quelque chose au sujet de l'isomorphisme de Curry-Howard avant que je ne pose ma question ? Oui ? Peut-être ... Je pense que la révolution qu'on est en train de vivre en physique est seulement une partie d'une révolution plus large, et il y a beaucoup de questions en physique dont les réponses utilisent la notion d'algorithme. Par exemple : quelles sont les lois de la nature ? Ce sont des algorithmes. Ainsi, il est possible de donner cette réponse, seulement parce qu'on a la notion d'algorithme. Et pendant des siècles, nous ne l'avons pas eue. Donc, à ces questions, on n'avait pas de réponses, ou des réponses ad-hoc. Par exemple : quelles sont les lois de la nature ? Compositions. Quand on avait des compositions, on n'avait pas les algorithmes. C'était un bon moyen d'y répondre. Et il y a énormément de domaines de savoir où sont posées beaucoup de questions auxquelles on a maintenant une réponse : c'est un algorithme. Et l'une des toutes premières questions au sujet desquelles on changea d'avis, fut la question : qu'est-ce qu'une preuve ? La réponse originale a été une séquence de formules vérifiant des règles de déduction et autres. Et commençant avec Kolmogorov – car derrière l'isomorphisme de Curry-Howard, il y a l'interprétation de Kolmogorov – il y a cette idée que les preuves, comme les lois de la nature, sont en fait des algorithmes. On a donc deux facettes, ou deux éléments d'une plus large révolution, et je pense qu'ils sont reliés de cette façon.

Maintenant considérons cette **Question** : le boson de Higgs existe-t-il ? J'imagine qu'aujourd'hui il y a des gens qui croient que le boson de Higgs existe, et des gens qui croient qu'il n'existe pas, mais chacun – vous pouvez prendre l'électron, si vous voulez (à la place du boson de Higgs) dont la plupart des gens, j'imagine, croient à l'existence – mais je pense que tout le monde est d'accord qu'on doit trouver une certaine sorte de procédure [des moyens de vérification pour une telle prédiction]. Ce peut être une expérimentation, ce peut être tout ce que vous voulez – dans ce cas – qui va permettre finalement, si on a assez de chance, de résoudre cette question. Et je serais très mal à l'aise si quelqu'un me disait que le boson de Higgs existe dans l'œil de l'expérimentateur, ou que le boson de Higgs ne doit pas exister car alors la théorie serait plus belle, ou si le boson de Higgs n'existe pas alors

c'est réellement un problème de l'Univers et pas le nôtre, et on peut continuer à postuler qu'il existe car ce serait plus joli. Donc je me demande si les deux questions dont on a (pas ?) discuté aujourd'hui ne seraient pas du même genre : on devrait essayer de chercher un genre de procédure pour répondre à la question, à un moment ou un autre. Un : l'univers est-il computable ? L'autre : l'univers est-il discret ? En un sens, ces deux questions ne sont-elles que des questions métaphysiques ou ces questions sont-elles associées à des expérimentations qu'on pourrait et qu'on devrait, en fait, continuer ?

Georges Johnson : Je suis désolé, nous n'allons pas avoir le temps d'engager là-dessus ... je veux remercier les orateurs. Peut-être pourrions-nous reprendre à l'occasion d'une prochaine conférence, et ce serait une raison supplémentaire pour l'envisager.

Fin de la transcription.

Annexe C

Open Discussion on *A Computable Universe*

Ce chapitre est une discussion des auteurs des diverses théories de modèles de calcul naturelle, digitales et algorithmiques du monde, parmi eux A. Bauer, T. Bolognesi, A. Cabello, C.S. Calude, L. De Mol, F. Doria, Ed Fredkin, C. Hewitt, M. Hutter, M. Margenstern, K. Svozil, M. Szudzik, C. Teuscher, S. Wolfram & H. Zenil]A. Bauer, T. Bolognesi, A. Cabello, C.S. Calude, L. De Mol, F. Doria, E. Fredkin, C. Hewitt, M. Hutter, M. Margenstern, K. Svozil, M. Szudzik, C. Teuscher, S. Wolfram & H. Zenil. Il s'agit notamment de questions posées par H. Zenil à ces auteurs. Une version en anglais a été publiée dans mon recueil (*A Computable Universe*, World Scientific, 2012).

Question from H. Zenil to M. Margenstern : In your paper (published in *A Computable Universe*) you provide the encoding of a very small Turing machine as an example of something that, written in the letters of DNA, is smaller than any possible virus. This seems to suggest that if Turing universality requires so little, there are good chances that viruses (if not as individuals then as a set) may be capable of universal computation and therefore capable of potentially any possible threat to any possible solution. Is this what you are suggesting? If this is the case, I wonder what your position is on the pervasiveness of universality in connection to the (un)pervasiveness of undecidability in nature, in particular in biological systems. It would seem that while one may find evidence for the pervasiveness of computational universality, one does not find many identifiable examples of or candidates for

natural undecidability in natural systems. Do you think there is any fundamental reason for this or is it that they have simply eluded our efforts to identify them? Charles Bennett, for example, enlightens us with a beautiful analogy which casts RNA polymerase as a specific purpose Turing machine, what amounts to a "truly chemical Turing machine" (from his "The Thermodynamics of Computation—a Review"), and more recently it has been suggested that the language of genes may have a computational power close to, if not the same as that of Turing completeness ("The language of genes" by David Searls in *Nature*). It seems we have very little to say about analogies with possible biological undecidability and how it may or may not play any role in biology. Would you have anything to say in this regard?

Answer by M. Margenstern : I think that the pervasiveness of universality is high and if it does not appear to be so, the reason might be that people have not tried to investigate biological phenomena with this idea in mind. What we can see depends on what we expect. If people do not care about universality or undecidability, many facts or features that would point to it cannot be identified. I do not think that there are absolutely objective conditions : we observe nature and this observation is not at all neutral. First, it may involve an interaction with nature, even if we try to avoid it. Second, the observation is highly dependent on the culture of the observer, in particular on his/her language, on his/her ideas, among them, his/her opinion on what he/she expects to see. Fortunately, some of us are able to see things which were not foreseen, but this is not that frequent and when it happens, it is not always accepted. This is another aspect of the social conditions in which science evolves and these conditions act on scientists, all the more so if they are convinced that they are immune to such influences. To go back to our subject, the studies about universality and undecidability are not known well enough outside computer science circles and very restricted circles of mathematicians and physicists. So it is not surprising that many things connected with universality have not yet been observed. Now, I would like to point out that some studies on ciliates seem to suggest the fact that the well known pointer and stack mechanism used in programming is known in nature. It seems that the explanation of some mechanisms observed in the reproduction of ciliates shows some evidence of this. There are several works by Gzegorz Rozenberg on this very topic. Now, with two stacks and a not very big automaton, we get universality. Furthermore, there are very interesting works by Eshel Ben Jacob on colonies of bacteria. This is what I consider in the paper I submitted for your volume. The remarkable ability of such colonies to adapt to extremely severe conditions is something which should be investigated. It might be possible that dialogs between universal

Turing machines could lead to super Turing problems. I really think that we should consider that recursively enumerable sets constitute the lowest level of life and that higher forms require us to go up in the hierarchy. Well, I have no proof of what I claim. It is just a direction of thought. I believe it is meaningful. The future will tell us whether or not this is the case.

Comment by M. Margenstern on C. Calude, F.W. Meyerstein and A. Salomaa's chapter : I am highly appreciative of the paper by Cristian Calude, Walter Meyerstein and Arto Salomaa entitled "The universe is lawless...". It is time to put an end to the dominion of the Big Bang theory. I do not understand how people using a mathematical model can claim that it expresses, even in its guidelines, the past of the universe. Mathematical theories make use of reversible time; how can they exhaust nature, which is not at all reversible? Let us note that in our discrete time of computations, time is irreversible : it is very often extremely difficult to run an algorithm backward. At the highest level of generality it is impossible. Let us note that the formal definition of sequential algorithms induces an irreversible time which is present in the very notations. As an example, an affectation is usually denoted as "var := expr;" and the notation " :=" explicitly indicates that what is on the left-hand side is not the same as what is on the right-hand side, and that there is a process, a consequence of which, after some time, is what we call an affectation. This is a very trivial example but we can see that in this way things are set on their feet and not upside-down. There are many things that could be said on this subject : how our mathematical theories do not give an intrinsic account of the notion of orientation (if you want to orient yourself in space you have to look at a watch; no mathematical theory can help you decide which orientation to choose between two possibilities), which is not the case in nature. So how can some people claim that our theories, which are an approximation of reality, allow us to define a time 0 from which everything starts? However, we can say that the present approximation constituted by mathematical theories is a very good one from a practical point of view. It is a rather convenient system for dealing with phenomena on earth and in space which is rather close to earth. Classical mechanics does not govern planets at the level of our solar system, as proved by the inaccuracy of predictions of the perihelia of Mercury made from within this framework. Well, this is not dramatic for planes in the air, ships on the seas, cars and trains on land. It is, however, interesting to note that GPS was rather recently made more efficient by computing the relativity effect of the speed of signals used to compute positions. It has not yet taken into account that the geometry of our space is probably not Euclidean. I also think that

taking into account algorithmic problems in the computability or incomputability of some mathematical models could lead to better approximations of reality. To me, the opinion that “the universe has local rules only” is very interesting and should enhance our ability to study it.

Question from H. Zenil to T. Bolognesi : You say that “We still lack experimental evidence for this conjecture,” the conjecture being “complexity in nature = emergence from computation”. It seems to me that the work of Wolfram and myself applying algorithmic probability as a theory of pattern generation from purely computational processes, based on the works of Levin and Solomonoff, is evidence in favour of the complexity in nature produced by computation, evidence that should not be completely overlooked even if it is not at all conclusive. Perhaps you were thinking of the smallest scales (which is likely since in your text you immediately turn to Planck units), where effectively we have no clue. On the contrary, there is an important discrepancy between the kind of complexity that can come out of quantum mechanics and the kind of complexity generated on the macroscopic level, which is deterministic in principle (as all the laws of physics are reversible).

So my question to you is what elevates a model, hypothesis or theory above the subjective, the aesthetic or the metaphorical, to use your own words? Is it their explanatory and predictive power? In which case, algorithmic information theory predicts the nature and frequency of certain patterns in the world, if the world were the result of computational processes. And while we don’t know whether nature is a computational process, we do have an idea what a world that is a computational process would look like, which gives us a basis for making comparisons (and even predictions). Its explanatory power is apparently underestimated but it easily explains why structure can emerge even out of almost nothing or out of randomness. Science has been working since the beginning on this very same assumption, that the world is structured and that one can derive universal physical laws from reality, but without knowing much about why this is so.

So I wonder what would make you feel that a computational theory of the universe is correct? We know that ‘computational’ doesn’t necessarily mean ‘predictable.’ What kind of verification tests could be conceived that would give them the status of a theory, would elevate them above fairy tales as models of the world?

Answer by T. Bolognesi :

You raise the fine point of predictability vs. irreducibility. The key element that discriminates scientific theories from other forms of knowledge, such as philosophy or religion, is the requirement of experimental validation, as taught by Galileo four centuries ago. I believe this fundamental will stay with us at least as long as the Aristotelian thinking that it has superseded. On the other hand, if the universe is animated by a computation, it seems reasonable to assume the latter to be irreducible, like that of elementary cellular automaton number 30, and this leads to the impossibility of predicting its evolution faster than in real time. However, I do not think that the consequences of this state of affairs are fatal for science. The ultimate, discrete ToE [Theory of Everything] (if any) should describe, with infinite precision, the evolution of the universe as a *whole*. Now, it is one thing to be unable to predict, in less than 10 years, what the whole universe will look like in 10 years — and emergent phenomena are indeed unpredictable, almost by definition, so that no scientific theory will ever be able to describe in advance, say, the future shapes that life will assume in our galaxy. It is another thing to be able to predict the exact mass of a few hundred elementary particles : the amount of information involved here is much smaller, and their derivation from a hypothetical universal computation, once and for all, is conceivable. That would constitute very convincing experimental evidence for the validity of the ‘theory’!

Although the most important test for the validity of a scientific theory is the prediction of measurable physical quantities, with the highest numeric precision, I believe that other, more qualitative forms of predictive power can also be considered, at least for acquiring some degree of confidence as to the validity of ‘an approach’, if not of a specific theory. For example, a computational theory that, starting from completely abstract mathematical/logical rules, yields the emergence of localized structures whose interactions bear analogies with those of known physical particles, should be regarded with more favor than one producing white noise all the time.

In a way, the brilliant idea that you mention — to spot clues to algorithmic behavior behind the distributions of empirical data — would also fit in this broader picture. The key observation at the core of that proposal seems to be that a random string looks random most of the time, while the output of a random Turing machine has greater chances of looking structured.

The big challenges I see in that approach, however, are : (i) how to reduce the immense variety of empirical data sets and associated probabilistic distributions available in the physical world to a single, representative empirical distribution; (ii) how to do the same in the computational world, where you may want subsystems to also interact, and where different layers

of emergence may each be characterized by its own laws.

Furthermore, there are other approaches that attempt to explain the emergence of structure in physical systems. One has been investigated for decades by Stuart Kauffman . In his networks of Boolean functions, ‘order for free’ is very likely to emerge — in the form of surprisingly small attractors in the configuration space — whenever the network satisfies some reasonable topological conditions. It would be interesting to further investigate the relations between the ‘order for free’ obtained by Kauffman’s generalized cellular automata, and that associated with Levin ’s miraculous distribution.

Question from T. Bolognesi to Ed Fredkin : As a premise, I wish to express complete support for your assumption that ‘space, time, state and all other properties and processes of physics are finite, discrete and deterministic’, and that the Discrete Theoretical Processes that ‘run’ the physical universe must be of a computational nature. The variety and complexity of phenomena that emerge from the computations of Cellular Automata [CA], or, in fact, of all Turing-universal models, have been widely explored in the last decades, from Zuse to Wolfram , and it is hard to believe that the strong analogies between the computational and the natural universe are purely accidental.

What I personally find a bit costly, however, is the assumption of a huge (infinite?) regular grid, or 3+1-dimensional Cartesian lattice, and associated program, as a prerequisite for starting the CA-like game of our universe. In order to justify the existence of this gigantic background structure, while avoiding the idea that it pops out from nothing, you suggest that there exists a sort of Super-Universe, that you call ‘Other’, in which such a grid is created, initialized, and started.

In doing so, you may have provided a more rational (and certainly original!) justification for the existence of the grid, but are you not simply moving the bootstrap problem one layer up, making it even harder? It might be that any cosmogonic theory will have to face some form of bootstrap dilemma, but one way to alleviate it is to stick to minimality criteria, and reduce as much as possible the gap between nothingness and the first instance of ‘something’ — the seed from which everything derives. Justifying the appearance of a Super-Universe from nothing seems more costly than assuming some simple initial condition for a universal computation. So, would it be possible/desirable to conceive the background grid itself, be it regular or not, as being incrementally produced, along with all the data that populate it, by the same universal computation that runs our familiar world, rather than

imagining it built in an ‘Other’ place, and perhaps even run by an actual, ‘Other’ Super Computer ?

Answer by Ed Fredkin : There are a number of observations that lead to thoughts about what I call ‘Other’. Our Universe has a number of characteristics that are essentially irreconcilable. There is a great deal of evidence that supports the ‘Big Bang’ theory of early events in the evolution of this universe. However, the idea that space, time, matter, energy and the laws of physics all appeared out of nothing is inconsistent with those same laws of physics. In addition, it defies common sense. One of the most fundamental of all laws of physics is reversibility. To believe some Big Bang cosmogonies, we have to also believe in the reverse process, where all of the matter and energy in the Universe collapse into a single point, and violating conservation laws, disappear into nothingness. It is also unreasonable to believe that the laws of physics existed independent of our physical Universe and somehow allowed for the origin of the Universe. Long ago, we had little choice in trying to explain the cosmogony problem ; but now that we are more familiar with theoretical computation there is the possibility of more sensible candidates for models of the cosmogony problem. Many current models seeking to invent solutions are one or another form of fairy tale. However some fairy tales are more reasonable than others. Given the assumption that the ultimate and correct model of the most microscopic processes in physics is some kind of discrete space-time-state system, then we can invent the most straightforward and reasonable fairy tale, the one with the fewest and the least outlandish assumptions. If we were to find that there are discrete space-time-state models for microscopic physics, then that could open the door to possibilities supported by common sense. Given an accurate, deterministic, discrete, space-time-state model of microscopic physics, it would be true that any computer could model a small volume of space-time and do it exactly. In a discussion that took place between myself and Marvin Minsky in approximately 1960, we agreed that if such a model, running on an ordinary computer, was the correct model, the computed process would not merely be a simulation of physics, but it would indeed be physics. The point is to consider the necessary requirements or conditions for Other to be able to host a computer, most likely a Cellular Automaton (CA), capable of simulating our Universe (if space, time and state are discrete). First, Other does not have to have the restriction that our Universe has of having a beginning or an ending. It does not have to have our laws of physics. It does not have to have our kind of space or space-time. Computation is so general as to put almost no restrictions on the nature of Other except for a lower bound on the number of states that had to have been committed for the simulation of our Universe.

The only requirement we put on Other is that it be able to host a CA that could run the process we call physics. The concepts of beginning and ending, which in one way or another seem to be a characteristic of our Universe, need not be a property of Other. There is no need for assuming that Other requires some kind of beginning. It may be hard for us to conceptualize worlds that do not conform to our physics, but it is clear that the demands made by the assertion that some system can support discrete computation are so minimal as to defy comprehension. Occam's Razor suggests that we look for simple explanations. A continually existing place— not in this Universe, not subject to the laws of our physics—that can create and run discrete processes capable of modeling this Universe may be the simplest explanation we will find.

Question from H. Zenil to T. Bolognesi :

When you talk about persistent structures in, for example, Wolfram 's Rule 110, as being analogous to physical elementary particles, structures in Rule 110 are supported by a completely regular background as you point out. Do you think this background, besides acting as a physical carrier for such structures to propagate, is also a necessary kind of ether supporting the computation itself? The question has to do with the way Turing machines are usually set to start out from a "blank configuration" and whether you think a blank configuration does or does not have a physical representation of computation in the real world. In the end, both Turing machines and cellular automata have, for example, their ultimate "physical" carriers—a tape for Turing machines and a grid for cellular automata—yet one can define both formalisms with pure functions without having recourse to any substratum. Could you elaborate on this in the context of a discrete space-time ? Does it mean that if discrete, the world can also be just mathematical ?

Answer by T. Bolognesi :

You mention (without necessarily sharing) the viewpoint that, since models of computation such as Turing machines and cellular automata can characterize the computable functions, we may directly refer to the latter for a fundamental theory of physics, and conclude that 'the world is purely mathematical'. I always feel 'uncomfortable' with this idea. The main reason is that, for a Turing machine to express a mathematical function, you need to provide it with a halting state, and to look at the final tape configuration only when the machine halts, while fully ignoring all the interesting things that have happened along the way.

On the contrary, for a Turing machine to express an evolving, artificial physical universe, you don't need to include halting states at all, and you concentrate instead on the 'shape' that the computation assumes in its detailed, step-by-step unfolding. In this respect, I totally support Fredkin's idea of a one-to-one correspondence between state and function in the physical world and state and function in the computational model : nothing in the computation should be deprived of a physical counterpart, and vice versa. This is one way to contrast the computational vs. the purely mathematical universe concepts : the physical universe (not the devil!) hides in the details—in every single detail of the computation, not in the final outcome.

You also raise the interesting point of the regular support, or substratum (cell tape or array) required by Turing machines and cellular automata for carrying out their computations, and of the periodic background that seems to be required for artificial computational particles to come into existence and play their games.

The main reason why I consider the two models mentioned only as useful *metaphors* for a computational theory of physics is precisely that they require this cumbersome, predefined support. I look with much more favor upon graph-rewriting models, such as trinet mobile automata (TMA), where the regular substratum is not assumed a priori, but can be expected to emerge from the computation itself, along with the particles that (may eventually) populate it, often starting from a tiny, two node graph. My experiments with TMA computations clearly indicate that many of the phenomena observed with cellular automata can also occur in the causal sets (intended as spacetime instances) obtained from this model, and the emergence of a regular background is certainly a most common one.

Question from C. Teuscher to M. Hutter :

A computational paradigm that does not allow for “programming” is pretty much useless. I wonder if a computational theory of everything must also include a “programmability of everything” in order to be useful?

Answer by M. Hutter :

All considered CToE candidates (q, s) are computable. A pair (q, s) is an algorithmic description of the observations of a subject in a universe algorithmically generated by q . Whether (q, s) is a correct CToE, i.e. correctly “explains” all observations, is demi-decidable by executing (q, s) .

Even the search process for better and better (i.e. smaller and smaller) CToEs itself is effective by standard program dovetailing.

And to top this, this process will find the best (absolutely shortest, given the evidence *o*) CToE in finite time.

What we cannot hope for is that this procedure halts thereafter (due to the halting problem). So we can never be sure we have found the best CToE, but likewise future observations have the potential to invalidate any (C)ToE found by any means.

Also in practice, more intelligent ways than brute-force search and simulation are of course used for finding good CToE candidates and checking their consistency with observations, which actually can still be a computational process.

I guess this constitutes what Christof Teuscher calls the “programmability of everything” (modulo computational limitations of course).

Question from C. Teuscher to F. Doria : Ideal analog computers don’t exist in nature because every physical quantity ultimately has a limited precision. How does your approach avoid that fundamental limitation to compute beyond the Turing limit ?

Answer by F. Doria : I’ve been trained as an engineer. Engineers build things, given some blueprint or some reasonable technical description. So, as the construction of an analog computer that would—ideally—settle undecidable stuff is within our reach, I want to build one and see how it performs and especially how it fails.

There are a few points I’d like to stress :

- The main theoretical difficulty when dealing with the halting problem are the nonstopping values (“infinite loops”) in the algorithm. But in actual circumstances we are only interested in a finite number of such values. It is known that in such a case there will always be an algorithm to do it, albeit a possibly very complex one.
- I wonder whether some real hypercomputer which performs well within some finite range can actually be used to speed up computations. That would be the main practical interest in building such a machine.

There are a few ideological (if I may say so) objections which deserve comment. For instance, “the universe is discrete and analog stuff are continuous.” One usually points out that Schrödinger’s picture of quantum mechanics deals with a smooth object, the wave function. And so on. But the possibility that the universe is essentially discrete wouldn’t affect the building of a practical machine, as I have suggested.

Why has nobody tried to build it and see how it performs? After all, the proof of the pudding is in the eating!

Question from H. Zenil to C. Calude : If the universe is lawful as you argue in your contribution to this volume, with the fair point you make about our models not fully matching nature, how could we account for the fact that many of these models actually capture much, if not all, the behaviour of many natural systems? The question could be rephrased by evoking Eugene Wigner’s claim about “the Unreasonable Effectiveness of Mathematics in the Natural Sciences”.

There is also the idea (popularised by Paul Davies, but originating with others, for example John A. Wheeler, if I’m not mistaken) that physical laws may change over time, just as physical constants may over the course of the history of the universe (so not all physicists embrace the idea that laws are unchangeable and true forever, even if most do). Of course our lifespan is so short that we are used to thinking of physical laws as perennial and immune to change, which is reasonable because were the laws to change we would not be able to make sense of the past in terms of the present (or make sense of the future). But assuming that laws may change, things like the inflation period of the universe or dark matter may find a reasonable alternative explanation (likewise the question of the tuning of the universe).

The beauty of the computational/algorithmic view from my vantage point is twofold. On the one hand there is the lesson learned from Turing’s work that data and programs are essentially not very different because one can always be encoded in the other. Hence there would be no need to separate them in a theory of laws+state. On the other hand, we find this beautiful convergence in computational power among different, initially unrelated models of computation, allowing us to discuss underlying general principles of physical reality under a single assumption, rather than having to posit different theories for different natural phenomena. In the final analysis science has been quite successful in merging phenomena that seemed at first to be unrelated, showing them to be governed by increasingly more general but accurate physical laws. Also, it seems to me that computational views have the advantage of not being contaminated by additional, apparently unnecessary assumptions, such as the presumption of the existence of real numbers, even if we cannot decide whether they exist or not by generating models because the answer only belongs to physics.

I think there is a distinction between whether the Universe is epistemologically or ontologically lawful. I think you are suggesting that the Universe

is *ontologically* “lawful”, but most of your arguments are for an *epistemologically* “lawful” universe. As you say, it is one thing to think there is no cause and another to think that the cause is unknown or unknowable. You propose that if the digital views are subject to the lawfulness arguments it is because they cannot be distinguished from algorithmically random strings. In other words you suggest that we may be embedded in an accidental orderly finite string of an infinite random sequence. Isn’t the effective probability of picking a regularity in an infinite Martin-Löf random sequence zero? Just like the chances of picking a rational number among all the real numbers ($\mathbb{R}$)? Doesn’t this introduce a problem tantamount to the apparent fine tuning of the universe?

Answer by C. Calude : The regularities and laws discovered and validated in science, in particular in physics, can be empirically demonstrated to have “local” validity, which is very impressive and useful. There is nothing mysterious or unreasonable in the fact that local regularities match natural phenomena well in a finite number of cases.

To put this in perspective, does the enormous empirical confirmation of the Riemann Hypothesis justify the statement “the Riemann Hypothesis is true”? Of course not!

Our point is that there is no such thing as a “law of the Universe” as long as we cannot prove it, and science cannot prove it. Why so? Because of the infinity which is intrinsic to any non-trivial law. For a better understanding of the nature of laws emerging from a scientific theory I quote the following paragraph from Deutsch’s book *The Beginning of Infinity* (Penguin Books, London, 2012, p. 4) :

...scientific theories are not ‘derived’ from anything. We do not read them in nature, nor does nature write them into us. They are guesses—bold conjectures. Human minds create them by rearranging, combining, altering and adding to existing ideas with the intention of improving upon them.

This is why it is no surprise that “physical laws may change over time”. A possible reading of science is as a history of failed theories.

“Isn’t the effective probability of picking a regularity in an infinite Martin-Löf random sequence zero?” The answer is negative. An infinite Martin-Löf random sequence is full of regularities : it avoids only a countable set of regularities (given by Martin-Löf randomness tests). For example, in every

infinite Martin-Löf random sequence any string appears with a very high regularity (Borel normality).

Understandably, scientists love their theories as parents love their kids. However, one should be careful about unreasonable love!

Reply by H. Zenil : I think *local validity* is opposed to the concept of a physical law as conceived by physicists. A physical law is supposed to be valid anywhere (and perhaps anytime) in the universe. Whether this is or is not the case would make for an interesting discussion, but I think the success of science and in particular of physics is in the derivation of more general laws explaining a greater number of physical phenomena (hence the replacement of old theories by new ones).

On the other hand, I thought that the definition of a ML-random number was that its digits were random in relation to any effective algorithm. In the best case only a finite number of bits are computable in a ML-random number, just as for a Chaitin Ω number. But even worse, because there are random sequences of arbitrarily high Turing degree it seems one would need an Oracle to spot the regularities. Is it that from the outside the probability of seeing these regularities is 0, though the metaphor that the universe is like a ML-random number is valid because we may already be in a regularity? My main point is therefore why, if it is so hard to find such regularities, if this is a ML-random universe, do we seem to do so so often?

Reply by C. Calude : An ML-random sequence is one which passes every ML-test of randomness, so only a countable number of tests. As there are an uncountable number of tests of randomness, it follows that any ML-random sequence doesn't pass an uncountable set of randomness tests. There is no sequence which passes all, even an uncountable set of, randomness tests, so in every sequence there are regularities, but these regularities are "invisible" to Turing machines.

A ML-random number avoids every c.e. regularity but has plenty of incomputable regularities. No scientific theory can find incomputable regularities, so it is lawless.

Reply by H. Zenil : The math makes sense, but then it does not accord with the metaphor that the universe may be like a ML-random number if we find (apparently computable) regularities in the world all the time (it seems we find more computable regularities than irregularities—or as you call them uncomputable regularities).

Is your answer then that we see regularities (1) by chance (2) because we construct them as regularities (3) because we see these regularities despite scientific theories (and algorithms), which then is a form of negation of the CT thesis? Does any of these or a combination of them represent your view?

Reply by C. Calude : We don't "see them" as the result of scientific investigation because they are incomputable. We only know they exist. If by some miracle we can "see" such a regularity, there is no way to prove it. A similar situation : I know there is a winning lotto ticket, but I cannot know which one is it ; the knowledge helps, maybe, by motivating me to play lotto, but it does not help me in any direct sense to get the winning ticket.

Question from M. Hutter to C. Calude : Common-sense laws are rules to govern behavior, are not necessarily causal, and do not enforce a unique/deterministic behavior. Precisely in this sense also the universe respects many laws and is very law abiding. This seems to disprove the lawlessness hypothesis.

The paper seems to hint at several hypotheses, or at least this was my impression of it. I try to formulate them below. I cannot say for sure whether the authors had any of them in mind. Some of them seem worth stating and discussing further. I would like the authors to indicate which of these hypotheses they subscribe to :

- (H1) physical laws are not causal [In my opinion, not worth discussing, since it's wrong]
- (H2) the causal laws could be re-interpreted in a non-causal way [has been investigated and possibly worth investigating further]
- (H3) some laws of physics are not causal [possibly worth investigating]
- (H4) our observable universe is an (accidentally) regular part of an (infinite) random universe/string. [this is reminiscent of what is known as the 'Boltzmann Brain' idea. The authors' concepts of disjunctive real numbers and lexicons seem reminiscent of the 'Library of Babel'. I have investigated the random universe idea and its problems in <http://dx.doi.org/10.3390/a3040329>.]
- (H5) Free Will undermines the causality/laws of our universe. [I'm skeptical but would be interested in hearing more on this]. In Section 6 I cannot find "Arguments in favour of the lawlessness hypothesis" for our physical universe, even when admitting the, in my opinion implausible, "free-will assumption". I actually subscribe to Footnote 6].

- (H6) The law that governs the universe is a random string, hence the universe itself is lawless. [Paraphrasing the authors. This seems wrong. Input (law) and output (universe) are not the same thing. Cf. H7.]
- (H7) Some aspects of the universe are random (like quantum mechanics), therefore the universe is lawless. [In my opinion, this is wrong. The output of most randomised algorithms exhibits some real, not just accidental, regularities]
- (H8) Some aspects of the universe are random, therefore there cannot be a Theory of Everything. [this is worth discussing and depends on what one requires from a ToE]

Answer by C. Calude :

If the “laws of the Universe” are considered in the following sense

Common-sense laws are rules to govern behavior, are not necessarily causal, and do not enforce a unique/deterministic behavior

then you can rightly conclude that

Precisely in this sense also the universe respects many laws and is very law abiding.

With this definition every common-sense regularity, no matter how deceptive it may be shown to be under a more careful scrutiny, qualifies as a ‘law of the Universe’. If we accept this description, then the Universe is lawful. However, I see too little justification to accept it.

I do not subscribe to any of the hypotheses (H1)–(H8), although, as you correctly observe, (H4) comes close to our hypothesis.

Reply by M. Hutter :

- (a) Do you believe that you/we can find or have found computable regularities in the part of the universe you/we observe?
- (b) Do you believe these regularities are locally valid, but not globally since the total universe is globally ML random?
- (c) Do you believe that the part of the universe you/we observe happens to be a regular sub-part of the total universe?

In case the answers to (a-c) are yes, your idea is just an ML version of the random universe with possibly the same problem and solution as discussed in <http://dx.doi.org/10.3390/a3040329>

The problem is essentially that it is a metaphysical rather than a scientific statement. If clearly expressed and marked as such, I have a-priori no problem with it, and find it even interesting to philosophize about, though I can't find any argument in favour of it in your paper.

So in question form :

- (d) Do you believe your lawlessness claim is a scientific statement ?
- (e) Do you regard your lawlessness claim only as a metaphysical statement ?
- (f) Do you believe your lawlessness claim is a mathematical statement ?

Here's some more context :

Since QM, most physicists do not subscribe to a deterministic universe anymore, since (QM) 'laws' do not determine the future uniquely and completely. It may sometimes still be presented like that, as e.g. in [28], if the difference is beside the point or the author is sloppy or ignorant or expresses a minority opinion. This of course undermines deterministic causality too.

- (g) Is all that you claim that the universe does not respect deterministic and/or causal laws ?

Note that an approximate law of the universe would still be useful and approximately valid. And there are ways to convert approximate laws into exact laws by coding the errors.

In Section 6 you (only) show that (a) in mathematics there is something like random numbers ; (b) (With QM) one can (apparently) produce physical random sequences. This does not exclude the fact that the universe respects stochastic laws, in the sense that there is a high probability that it has certain properties(=laws?) (e.g. that the relative frequency of 1s in the above QM sequence converges to a limit).

Then you write in Section 8 that "the hypothesis can be experimentally illustrated and tested ... [12]" But [12] only investigates whether QM produces "true/ML" randomness. Unless you restrict "laws" to the deterministic, it doesn't increase the plausibility of your hypothesis

(another indirect indication that you reject stochastic laws for our universe). You seem to jump quite quickly (and, in my opinion, unjustly for several reasons) from "there is true randomness in the universe" to "the universe is lawless".

Final reply by C. Calude : The lawless hypothesis is the negation of the current dominant hypothesis that the universe is lawful; it should be analysed in this context. Given the limits of this discussion, my previous

answers together with the independent comments of Maurice Margenstern say it all.

Question from A. Cabello to C. Calude and K. Svozil : About using quantum systems protected by value indefiniteness grounded in the Kochen-Specker theorem for generating random bits (p. 8 of their contribution) : There are many ways in which a theory can have value indefiniteness (or contextuality of results), and quantum theory has a very peculiar form of value indefiniteness (which, incidentally, suggests that the universe is not lawless). By that I mean the following : A standard way to experimentally reveal nature's value indefiniteness (predicted by quantum theory) is through the violation of Bell inequalities. However, there are two different ways in which quantum theory may violate a Bell inequality : There are type-1 scenarios in which the maximum quantum violation is not the largest violation achievable by the no-signalling principle, for example in a Bell inequality with two parties, each with two settings, each with two possible outcomes [S. Popescu and D. Rohrlich, *Phys. Lett. A* 166, 293 (1992)]. In these situations, the observed correlations still can be expressed as a fraction of classical correlations (in which values may be defined) and a fraction of nonclassical ones [A. Elitzur, S. Popescu and D. Rohrlich, *Phys. Lett. A* 162, 25 (1992)]. In contrast, there are type-2 scenarios in which the maximum quantum violation is equal to the largest violation achievable by the no-signalling principle, and the observed correlations are fully nonclassical, for example in a Bell inequality with two parties, each with three settings, each with four possible outcomes [L. Aolita et al. *Phys. Rev. A* 85, 032107 (2012)]. The question is whether, for generating randomness, there is any difference between these two scenarios.

Answer from C. Calude and K. Svozil : Adan Cabello rightly states that there are many ways in which a theory can have value indefiniteness or contextuality of results. We would like to stress that what is presently understood as *value indefiniteness*—the dependence of an individual (non)observed observable or element of physical reality on what is measured alongside it (its “context”)—might be sufficient, but certainly not necessary, to interpret Kochen-Specker type theorems. Indeed, we consider this type of “explanation” very much inspired by the old quasi-classical realistic view of physics ; and it is no wonder that John Bell, a staunch realist, is often quoted suggesting that [J. S. Bell, *Reviews of Modern Physics*, 38, 447–452 (1966)] “the result of an observation may reasonably depend . . . on the complete disposition of the apparatus.”

We also agree with Adan on the distinction between, as he denotes them, Type-1 and Type-2 scenarios. For a detailed analysis, see also K. Svozil, *Natural Computing* 10(4), 1371–1382 (2011) and K. Svozil, *Natural Computing* (Online First 2012), DOI : 10.1007/s11047-012-9318-9].

C. Calude aside comment to A. Cabello : In an aside you point out that “value indefiniteness ... incidentally suggests that the universe is not lawless”. I disagree. We have to distinguish between the theoretical expressions of value indefiniteness (the Kochen-Specker theorem, for example), which are mathematical statements, and the **claim** that these statements are universally valid in the quantum universe. The **claim** can only be tested a finite number of times in contrast with the intrinsic infinite nature of each of these mathematical statements, so according to the arguments of our paper “The Universe is Lawless...”, the law of value indefiniteness is in the same class as all known physical laws : they cannot be proved, so they are not universal.

Question from C. Teuscher to C. Calude and K. Svozil : Most natural machinery, such as a biological neuron, is not universal but processes information in a highly specialised way, in which Turing universality is irrelevant. So why should one care about Turing’s barrier for practical/natural applications when most of them don’t even require Turing universality ?

Answer by C. Calude and K. Svozil : Turing’s barrier of universal computation, and the physical Church-Turing thesis in general, are very relevant metaphors of what actually—or rather, in the limit of unbounded space and time—could be achieved by physical and biological systems, even though many (bio)physical systems are specialised and incapable of performing universal computation.

More generally, one should not confuse necessity with sufficiency. Pointedly stated : although a mobile phone may not easily be seduced to compile a LaTeX document such as this one, there exist many physically realisable computers which can handle all kinds of requests, including mobile phone simulation and LaTeX computation, and a lot else besides.

This does not diminish the value of tiny mobile phones, or of desktop computers ; it is just that the latter can do “more” than the former. There is Turing’s limit with regards to simulation, and that Turing limit falls short of allowing quantum random numbers. But of course, very specialised systems are also incapable of “producing” incomputability through computability.

Finally, universal machines are marred by undecidable problems, but sometimes non-universal machines are not. Distinguishing between universality and non-universality helps us know how decidable a model is. For example, most parts of Algorithmic Information Theory are undecidable (because of universality) ; finite-state complexity retains many features of that theory but is decidable, so it has a greater chance of being practically applicable (see C. S. Calude, A. Salomaa, T. K. Roblot. Finite state complexity, *Theoretical Computer Science* 412 (2011), 5668–5677).

Question from C. Teuscher to A. Cabello :

What if it turns out that the universe does not compute anything useful ? In other words, why should we care what kind of computer the universe actually is ?

Answer by A. Cabello : The universe computes itself. It computes the present and the future, you and me, our life and death. It also produces individuals who do not care too much about whether all these things are “useful”, but definitely care about how the universe is working.

Question from H. Zenil to A. Cabello : For the Bell experiment to validate quantum reality one would need to design it to cover all possible open ends or loopholes, as you explain. Your paper seems to suggest that the only open loophole today is the problem of measuring fast enough to guarantee that measurements cannot influence each other and thus account for the correlation. It seems to me that your result tends toward ruling out the possibility that the universe is a classical computer, using an argument based on the amount of resources needed to keep track of the computation before overloading the space with enough information to keep producing the correlations that we see after the quantum experiments. Does this rule out the possibility of a completely deterministic universe, where the correlation would be explained simply because neither the particles nor the observers are independent of each other ? And is a full deterministic universe considered to be a hidden variable theory ?

On the other hand, all tests of Bell’s inequality seem to assume that one has the freedom to choose the detectors’ settings and that such freedom has to be indeterministic for the interpretation of the results of the experiment to work. The so-called free will theorem, for example, seems to be valid exactly in its reverse, negative version, that is, that if one does not have free will

then particles do not. Free will has been connected to randomness since the beginning, but it has played the role of devil's advocate, where sometimes it is the cause of randomness and sometimes it is the consequence of randomness.

Answer by A. Cabello :

Yes, a deterministic universe would be definitely a hidden variable model. Without making assumptions, I see no way to rule out a completely deterministic universe. However, we can rule some partially deterministic universes out if we make extra assumptions. The assumptions in Bell's theorem are : (i) Outcomes determinism, (ii) bounded speed of information, and (iii) observers' free will. The assumptions in my approach are (i), (ii') bounded density of information, and (iii).

I do not think that there is evidence to support the negative version of the free-will theorem. The free-will theorem states that, if the observer has free-will to choose the experiment on the particle, then the particle also has a certain amount of free-will in producing the corresponding outcome. The free-will theorem emphasizes the connection between (1) the assumption of the observer's free-will and (2) the conclusion that the outcome cannot be determined. However, the fact that (1) cannot be tested does not necessarily imply that (2) is wrong. Interestingly, the quantum protocols for randomness expansion in which the input is a private random string (the observers' free-will) and the output is a longer private random string, suggest that particles' free-will is greater than observers' free-will.

Question from H. Zenil to A. Cabello :

Stephen Wolfram has suggested in the past that pseudo-randomness (the only possible kind of randomness in a fully deterministic universe) has the obvious particularity that when its generating process is repeated with the same seed, the result is always exactly the same. I see you tackle the question of repeatable and compatible experiments. Could you elaborate a bit on how this idea of testing the possible pseudo-random nature of the universe (i.e. the possible assumption of being carried out by a classical computer) is connected to your paper? At a first glance, one may think that Bell's experiment yields the correlation because one is restarting the pseudo-random generator with the same seed. Is this what the violation of the Holevo bound actually rules out, on the grounds that a classical computer lacks the storage capacity to keep up with the information needed to simulate such correlations? Would your result still hold if the universe was deterministic from the inception?

Answer by A. Cabello : No. The idea is that, if a supposedly simple physical system acts as a classical computer, then it must store a large amount of information in order to provide outcomes in agreement with the rules of the game (quantum mechanics). If you assume that the system is actually simple, then it cannot act as a classical computer. The conclusion is based on the assumption that the system must store all the information needed to consistently answer any possible question. If the questions are determined, then no large memory is needed.

Question from H. Zenil to A. Cabello :

Quantum computation takes advantage of the physical effects of superposition and entanglement, leading to a new paradigm. In quantum mechanical computation models, all events occur by unitary transformations, which apparently means that all quantum gates are reversible. But it is not clear whether standard quantum computation fully captures quantum mechanics or is already committed to a specific interpretation of QM. This is because in some sense the standard interpretation of QM suggests that quantum processes are indeterministic, but if quantum gates in quantum computation are reversible, this clearly suggests the opposite, viz. that quantum processes are also deterministic (as I think is the view of Feynman, Bennett and Deutsch, to whom we owe the theoretical basis of the quantum computer). So either the standard model of quantum computation does not capture all the essential properties of quantum mechanics according to the mainstream interpretation, or quantum mechanics, just like classical mechanics, is reversible and therefore in a strict (perhaps not practical) sense, deterministic. Could you shed some light on this subject ?

On the other hand, in standard quantum mechanics, decoherence implies that the system considered is not closed, but interacts with the environment. Thus, while there is a leak of quantum information from the system into the environment, information is still conserved in the embedding universe. Is there any loss of information during decoherence ? Is it the case that the possibility of a fundamental information loss cannot be completely ruled out and is explored in alternative theories ? Are there any phenomena that we know of that imply a real loss of information ?

Answer by A. Cabello : It is not true that in quantum computation “all events occur by unitary evolutions”. Usually quantum computations consist of unitary evolutions followed by measurements that have intrinsically unpredictable outcomes. Quantum gates are reversible, quantum measurements are not.

I would need a more precise definition of what you mean by a “fundamental information loss”. In any case, let me point out that the quantum information degraded under decoherence is the information an observer has about the probabilities of the different possible outcomes of any possible experiment involving specific physical degrees of freedom.

Reply by H. Zenil : By loss of information I mean information that cannot be recovered in the logical sense, e.g. in non-reversible computation. Here is an interesting quote concerning an additional loophole to close, viz. a local “leak” of information—besides the possibly deterministic nature of quantum mechanics—in order to perhaps avoid some of the problems you have pointed.

In his *Horizons* paper (<http://arxiv.org/pdf/gr-qc/0401027v1.pdf>) G. t’Hooft writes :

The most important problem of the ones just alluded to is that deterministic evolution seems to be difficult to reconcile with a Hamiltonian that is bounded from below. It is absolutely essential for Quantum mechanics to have a lowest energy state, i.e., a vacuum state. Now the most likely way this problem can perhaps be addressed is to assume not only deterministic evolution, but also *local information loss*. As stated, information loss is difficult to avoid in black holes, in particular when they are classical. It now seems that this indeed may turn out to be an essential ingredient for understanding the quantum nature of this world. (emphasis in original)

This sends the author to examples of irreversible cellular automata in Wolfram ’s NKS :

Simple examples of universes with information loss can be modeled on a computer as cellular automata.

So in the case of QM, I understand your claim is that loss (or degradation) of information in this reversible sense is at the level of the observer making a measurement, meaning that if one tries to reverse the operation (entangling the particle again) and make another measurement, the second measurement is disconnected from the first one, and there is no way to recover the first one from the second measurement. Is this the kind of irreversibility and loss of information implied by quantum mechanics? Is this, then, a “fundamental loss of information”? As opposed to, for example, classical mechanics,

where apparently irreversible processes release heat from where one can, in principle, recover all the information and reverse any classical process.

Question from H. Zenil to C. Teuscher :

Wolfram has pointed out that there is an almost embarrassing difference between engineered artefacts, which tend to be simple and predictable yet are not very robust, and are often faulty compared to things easily identified and produced by nature, which are more complex and, more importantly, robust, even if unpredictable in almost every respect. I think you are saying something very close, making your case in favour of an unconventional model of “unorganised” computation (an alternative to von Neumann’s successful but apparently limited model). One property of computational systems is that every time one runs a program for a given input the output is exactly the same. Deterministic behaviour is a fundamental property of classical computation (even if this doesn’t mean that computation is predictable). My question is whether you think that running a program on an unorganised model of computation will also turn out to be even less predictable, and how negatively or positively this may impact the practical use of such a computational paradigm. Experience may suggest that, for example, natural or living systems are difficult to program (to make them behave the same way in the face of the same stimuli).

The second part of my question concerns the mechanism of random self-assembly of complex networks. Could you elaborate on how, for example, small world networks can self-assemble? By what mechanism? The paper suggests you have run tests using evolutionary algorithms. By what other means or processes might we be capable of producing complex networks other than by engineering the network in advance?

Answer by C. Teuscher :

Many of the computational substrates and computer models we proposed are disorganized and/or exhibit highly stochastic behavior. While regularity and deterministic behavior are desirable properties of most classical computational models, they mostly serve the purpose of making the designer's and programmer's lives easier. The approach we took instead was to harness what we could easily and cheaply build on a massive scale. It turns out that such things are rarely perfect and regular. But irregularity and unpredictability don't mean that the computation will necessarily be unpredictable too. With further scaling down CMOS, most of the "classical devices" will have high variation and become increasingly stochastic in their behavior because of fundamental physical limits (e.g., thermal noise). However, we know how to build abstraction layers in traditional CMOS design that hide such behavior. Perfect computation based on imperfect components is something we do often with classical models too. Of course each abstraction layer typically comes at a certain cost in terms of overhead. Nothing prevents us from building such abstraction layers with unconventional models as well. However, the real question is whether that should be the goal. For example, do we really want to compute binary logic with a reaction-diffusion system that can solve PDEs directly? So in that sense, I think unconventional substrates need unconventional computer paradigms and unconventional programming. What works well for CMOS doesn't necessarily work well for unorganized machines. I think nature provides us with many examples where the physical/chemical substrate is harnessed in a very particular way that is unique to that particular substrate.

It's been well known that small-world networks evolve if you consider both the cost and the performance. Such networks simply occupy a space in the entire space of networks that optimize for these trade-offs. That's the main reason why many natural networks are small-world. E.g., the neural network of the brain needs to perform well but is also subjected to cost (e.g., energy, space) constraints. We have hypothesized in the past that certain ways of growing nanowires (e.g., from seed points) will lead to small-world-like networks because the wires "compete" for limited resources as well (e.g., space, chemicals). Preliminary experiments in collaboration with Los Alamos National Lab have shown that we indeed get wire-length distributions that can lead to small-world networks. However, more experiments are required and the results may not apply to all growth methods and self-assembly approaches.

Question from F. Doria to H. Zenil : What is your opinion on the issue

of hypercomputation? Is it feasible? Impossible?

Answer by H. Zenil : I was once myself a hypercomputation scholar (the title of my Master’s thesis—in French—was “Hyper calcul”). Paradoxically it wasn’t until I had a good knowledge of it that I started to better appreciate and to increasingly enjoy the beauty of the digital (Turing) model. One powerful reason is that hypercomputational models do not converge in computational power. There are a plethora of possible models of hypercomputation while there is only one digital model (in terms of computational power). I find it astonishing how little it takes to reach the full power of (Turing) universality, and I don’t see the power of universal machines as being subject to any limitation. Quite the contrary. This shift occurred during the time I started working for Stephen Wolfram , who certainly may have influenced me. I happen to think the question has a truth value and is ultimately susceptible of a physical answer. However, I don’t think the answer will emerge in the foreseeable future, if it ever does.

Given all this, I find your contribution to the subject interesting in that you point out a particular (“ideal”) model of hypercomputation that I think gets around one of Davis’ objections, though it doesn’t fully address the problem of verification. Unlike a Turing computation, which can in principle be verified by carrying it out by hand, step by step, the inner working of a hypercomputer can only be followed by another hypercomputer. The caricature version of the problem is in the whimsical answer given by the (hyper?)computer Deep Thought , which proposed “42” as the “Ultimate (uncomputable?) answer to the Ultimate Question of Life, The Universe, and Everything” [parenthesis added] in *The Hitchhiker’s Guide to the Galaxy* by Douglas Adams. The only way to verify such an answer would be by building another, more powerful and even less understandable computer.

This makes me wonder whether we ought not to favour meaningful computation over what could potentially be hypercomputation, even if hypercomputation were possible. There is a strong analogy to the concept of proof in math. Mathematical proofs can be of two types. They either serve to convince you of the truth of a statement you weren’t certain was true, or else to provide logical evidence that a statement intuitively believed to be true was in fact so (e.g. the normality of π). But in the latter case, why would one bother to provide a proof of a statement that nobody would argue to be false? It is because ideally math proofs should provide insight into why a statement is true, and not simply establish whether or not it is so.

I also think, unlike a few other authors, that taking Oracle machines—proposed by Turing—as hypercomputers is mistaken and misleading. The

question as regards hypercomputation is whether one thinks Oracle machines are actually realisable, and I think Turing had, at the very least, no ambivalence on the subject. Both his work and his later philosophical positions indicate that he certainly never thought that Oracle machines would ever be constructed, only used for the purpose of studying the limits of computation and relativising the concept of computation into computability degrees which told us something very interesting, equivalent to Gödel's result, that even assuming an Oracle one would still find problems of increasingly greater difficulty than even Oracles are able to solve.

The study of infinite objects has, however, given us great insight into profound and legitimate questions of math and computation, such as the question of the nature of number and computation. And it has been immensely useful to focus on limits in order to better understand these concepts. As De Mol has concluded in her contribution to this volume (“Generating, Solving and the Mathematics of Homo Sapiens : Emil Post’s Views on Computation”), it is at the boundary between decidability and undecidability that one seems best positioned to answer the foundational question of computation. And there are some examples where the study of a hypercomputational model is more valuable—from my personal point of view—not as a model per se but because of its ancillary results (see e.g. José Félix Costa’s paper “The ARNN model relativises $P = NP$ and $P \neq NP$ ”, 2011). Unlike most people, I think the contribution of Siegelmann and her ARNNs has, for example, much more to say about computational complexity and therefore classical computation than about hypercomputation (Siegelmann’s ARNNs model has come to be strangely revered among hypercomputation enthusiasts). As Doria points out in his contribution to this volume (Blueprint for a Hypercomputer), Martin Davis provides (in “The Myth of Hypercomputation”) some criticisms (some more justified than others from my perspective : I agree with Doria that the problem of verification of a computation is not necessarily an argument against hypercomputation but whether hypercomputation is legitimate is a true problem). Among Davis’ criticisms there is the fact that the ARNN model is circular and its value as a hypercomputational model depends ultimately on physics (which is something we already knew, i.e. the ability to encode non-computable numbers as weights in an artificial neural network), and the model is by no means a model that can suggest that building such a neural network or a hypercomputer is possible.

Question from H. Zenil to L. De Mol :

Proving the undecidability of the halting problem and providing an answer to Hilbert’s program is the extraordinary negative result of Turing’s

seminal work. The positive result of Turing's paper is, however, the concept of computational universality and the proof and construction of a universal (Turing) machine. Judging from Post's theses, which are not only equivalent to but anticipate those of Church and Turing, computational universality seems to be implied in his work, as indeed in his quest to find a general algorithm for all mathematics in the context of, I assume, the Entscheidungs problem. I wonder, however, about the weight that Post gave to the construction of the concept of universal computation, and whether this had an impact on the way the concept was or was not conveyed in his work, and hence on the difference in the reception of his work as compared to Turing's. Could you please elaborate on this and explain how Post might have realised and weighed the value of such a construction? As you say, even his positive thesis (Post's second) seems more profoundly rooted in the limits rather than the possibilities, in this case of a universal constructor (to use a term used by von Neumann).

Answer by L. De Mol : Post's 1936 note certainly does not contain the description of a universal computer, since it is but a short note. Also, in the 20s Post most probably did not have a detailed description of a universal computer (or something equivalent like a universal tag system or normal system) but one does find traces of the idea in his 'Account of an anticipation' of his work in the 20s, where it plays a role comparable to Turing's universal machine. Two (theorems)¹ in this manuscript mention the complete normal system K which is so called "*because, in a way, it contains all normal systems*" and which is compared with Turing's universal computing machine in footnote 95.² The first (theorem) states the existence of K , the second, as Post explains in footnote 99, "*shows [when combined with Post's thesis I] not only that the finiteness problem [the decision problem] for the class of all normal systems is unsolvable, but that the same is true for one particular one of them.*" Of course, since this work was not published before Turing's 1936 paper (the 'Account of an anticipation' was posthumously published in 1965 by Martin Davis), it was not received in the same way as Turing's paper. Later, in Post's influential *Recursively enumerable sets of positive integers and their decision problems* the notion of a complete set K is introduced which is similar to that of the complete normal system K . Several theorems are proven in that paper for K . However this paper was published only nine

1. Post puts some of the theorems of his 'Account of an anticipation' within brackets because he didn't have detailed proofs but merely sketches for those theorems

2. These footnotes were later added by Post when he was preparing the manuscript for submission to the *American Journal of Mathematics* in 1941.

years after Turing's 1936 paper and worked with a formalism that is far less 'intuitively appealing' than Turing's machines.

Question from H. Zenil to L. De Mol :

The general sentiment is that Turing's model was closer to the mechanical realisation of a mechanical machine, while Church's was at the opposite extreme, closer to what we today recognise as a computer language, hence software, while Post's model is often regarded as lying in between. Your paper has convinced me, however, that Post's was closer to Turing's mechanical realisation than to Church's, hence not exactly in an intermediate position. Post also seems to have been closer to a concept of complexity than any of the other proposers, as it apparently arises quickly from running even the simplest of Post's rule systems. Is this something you suggest or would agree with?

Answer by L. De Mol :

It depends on which model you are talking about. His formulation 1 is almost identical to the Turing machine and in this sense can hardly be called intermediate. However, it is interesting to point out that Turing's formulation is in terms of 'hardware', whereas Post's formulation is more in terms of 'software', giving a description in terms of sets of directions in a well-defined language. Post's normal systems, his canonical form C and tag systems are very different from a Turing machine and perhaps closer in spirit to programming languages than Formulation 1. In fact, it seems that John Backus was inspired by Post's production systems when he developed his normal form, calling them '*just the thing*'³ and Patterson of the Burroughs company invited Post to the Burroughs research division because "[your production systems and theory of canonical languages] appeal to us as possibly the most natural approach to the theory of computability from the standpoint of one interested in syntactical machines." (Patterson in a letter to Post, Dec. 17, 1952). It might well be noted that Post's production systems also had an important influence on the development of formal grammars, his work being an important source in Chomsky's foundational papers.⁴ Post was clearly

3. It should be noted though that it is not entirely clear where Backus knew Post's work from (See Martin Davis' paper *Influences of Mathematical Logic on Computer Science* published in R. Herken, *The universal Turing machine : A half-century survey*, Springer, 1995).

4. For more information on Post's influence on computer science, see Martin Davis, Emil Post's contributions to computer science, in : *Proceedings of the fourth annual symposium on Logic in Computer science*, June 5-8, 1989, pp. 134-136.

very aware of the possibility that simple programs can give rise to complex behavior. It was during his research on tag systems (which he called primitive forms of mathematics) that he understood that one does not necessarily need complicated machinery to have complex behavior (he described the behavior of some tag systems as being of ‘bewildering complexity’). Most probably guided by this insight, Post then proved that the “*meager formal apparatus of [the] normal systems can wipe out all of the additional vastly greater complexities of the canonical form*”. This is his normal form theorem, a theorem which led him to believe that in fact the whole of Principia Mathematica could be reduced to the normal form. He did not have a detailed proof of this at that time (the reduction of PM to normal form). This is the reason why Post did not anticipate the negative solution of the Entscheidungs problem for the restricted functional calculus proved by Church and Turing in 1936.

Question from H. Zenil to L. De Mol :

I notice that several of the papers in this volume mention the fact that math seems grounded in historical accidents (the reason why Post insisted that Church’s thesis and equivalent formulations should be treated as hypotheses). In Post’s words “I consider mathematics as a product of the human mind, not as absolute”. Just as Bauer seems to suggest in his contribution to this volume, and Sutner, in the problem he has with unnatural intermediate degrees of computation. Do you have any position in this regard? How accidental is the history of computer science? And how accidental it is compared to, say, the history of mathematics?

Answer by L. De Mol :

I do not really understand why the idea that math is a product of the human mind somehow implies that math is grounded in historical accidents. Of course, if one understands math as a product of the human mind then one can hardly deny that it is rooted in a history and, hence, that it is subject to history’s unpredictable and erratic ways. To call certain events in the history of computer science or mathematics ‘accidental’ is tricky since this depends on the theoretical notion of randomness one is using. Furthermore, it is not certain whether true randomness exists at all in nature, let alone, in human history. Notwithstanding these theoretical points, I think that events in the history of math or computer science can surely have a certain amount of accidentality or, at least, be the result of some exceptional or particular combination of events, but that does not mean that a given result

from that history would never have been, had that particular chain of events not occurred. For instance, one can wonder whether the design of modern computers would today be named after von Neumann if not for Goldstine's chance meeting with him at the Aberdeen railway station, or if Goldstine had not distributed the so-called First draft of a report on the EDVAC design as he did. But this does not mean that the electronic and programmable computer would never have been invented. It only means that it might have been named after someone else and that part of its history would have been slightly different. I think that one would need some notion of robustness to discuss the 'accidentality' and non-accidentality of history in a more serious manner. The difference between the history of mathematics vs. the history of computer science is probably that computer science is much more guided by/dependent on the history of technology, industry, economy and politics than mathematics. In that sense, one could say that the history of computer science is more 'accidental' than that of math but only insofar as it is more dependent on the developments of other histories, not in some absolute way.

Question from H. Zenil to K. Sutner : The lack of correspondence between the abstract and the physical world seems sometimes to suggest that there are profound incompatibilities between what can be thought and what has meaning in the real world. One can ask, for example, how often one faces undecidable problems. And it seems the case that in the practice of advanced math we have managed to avoid undecidability (even though this set is a natural computational degree). One may think that long lasting open mathematical problems are natural candidates for undecidable propositions. However, the history of math suggests that it is a matter of time before someone with some theory provides a proof for every current open mathematical problem. In the same way that an ad hoc proof was tailored to give a positive answer to Post's problem, could it be the case that math gets around to undecidability by producing proofs in the form of more powerful theories than the theories in which the original problems were stated? As we know this is the case with a simple mathematical statement such as Fermat's last theorem, proven by Wiles with sophisticated mathematical artillery. Do you have any comments on this? How natural/unnatural and common/uncommon a practice is this?

Answer by K. Sutner :

One has to distinguish between undecidability in the sense of classical recursion theory and undecidability in the sense of independence within the framework of a particular mathematical theory. For example, the problem

of finding integral solutions of Diophantine equations is undecidable in the computational sense. This will come as no surprise to anyone who has studied Diophantine equations; it is exceedingly difficult to handle even polynomials of rather low degree. The fact that there is no algorithm that can determine the solvability of Diophantine equations can be established within any standard formal system of number theory, such as Peano arithmetic. On the other hand, the question of whether every set of reals is measurable cannot be answered in the context of a much stronger system such as Zermelo-Fraenkel set theory, nowadays the default background theory for mathematics, thanks in no small part to Bourbaki's efforts more than half a century ago. One could argue that, given the increasing importance of computation, set theory is no longer the preferred candidate for that role, but we digress. If we accept the axiom of choice, there are non-measurable sets. Worse, by a famous theorem due to Banach and Tarski, there are paradoxical decompositions of a sphere that can be reassembled to produce a sphere twice the size of the original one. On the other hand, Solovay has shown that one can construct a model of Zermelo-Fraenkel set theory where every set of reals is measurable, assuming there are inaccessible cardinals. Thus, the existence of non-measurable sets of reals is independent of Zermelo-Fraenkel set theory; it cannot be decided within this context and it is not entirely clear how one should augment the standard system to settle this particular question. Most mathematicians would seem to favor adopting the axiom of choice, paradoxical side-effects notwithstanding.

As to long-standing open problems, the Riemann hypothesis is something of a stand-out. From the logical perspective, the statement of the Riemann hypothesis requires no more than universal quantification, so it is fairly simple in that sense and comparable to Fermat's conjecture. A lot of computational evidence points towards the correctness of the hypothesis. Yet it is not unreasonable to assume that a proof would use tools outside of the scope of Peano arithmetic, much as Wiles's proof did. Of course, a counterexample could be much simpler in nature. This is a bit different from the solution to Post's problem, which is irritatingly artificial but can be established within the confines of number theory. Or consider what is currently arguably the most important open problem in theoretical computer science, the P versus NP question. Here it is really unclear how much machinery is needed to deal with the problem. There may yet be a simple polynomial time algorithm that determines satisfiability of Boolean formulas—bear in mind the recent discovery of a polynomial time primality test that uses in essence no more than high-school algebra. Or we might be able to construct some analogue of the Friedberg-Muchnik argument that establishes the existence of a problem

intermediate between P and NP. This approach is somewhat less promising since any such argument would presumably be invariant under oracles, and we know that this is not the case. Lastly, the problem might be independent from some standard axiom system like ZFC. For example, it might be the case that there is a polynomial algorithm for satisfiability, but we cannot establish this fact within the given framework and we would need to propose some plausible extension to settle the question. Of course, the issue of plausibility here is quite problematic in itself. But that should not be interpreted as a fundamental flaw in the method; ever since Gödel demolished Hilbert's program in its original form we have known that there is no hope of reducing all of mathematics to a simple, finitistic basis. It seems to me that thoughtfully constructed "sophisticated artillery" is both necessary and quite acceptable from a foundational perspective. As Hilbert said, "We must know, we will know."

Question from Bolognesi to Hewitt : The computational universe conjecture, in one of its extreme forms, suggests that the physical universe is a manifestation of the emergent properties of a computation, one possibly deriving from the execution of a simple program started from simple initial conditions. Wolfram's elementary cellular automata are a popular metaphor for this, but Turing machines and various other Turing-complete models have been investigated under this perspective. One way to partially abstract from the specific features of one or the other model is to focus on the causal structure of computation events—a structure that could then be taken as a discrete model of spacetime.

In your paper you raise the point that the Actor Model is stronger than Turing machines, using an 'unbounded nondeterminism' argument. I do not know whether you have specific opinions on the idea of understanding Nature quite literally as computation. If so, I would be interested in knowing your opinion on the role that the Actor Model could play in that context, also in light of your claim that the model was inspired by the laws of physics. In particular, causal structures (partial orders) of events would seem to derive quite naturally from actor system computations...

Answer by C. Hewitt : The Actor model is largely compatible with the views of [Rovelli 2008] : a pen on my table has information because it points in this or that direction. We do not need a human being, a cat, or a computer, to make use of this notion of information.

Relational physics takes the following view [Laudisa and Rovelli 2008] :

- Relational physics discards the notions of absolute state of a system and absolute properties and values of its physical quantities.
- State and physical quantities refer always to the interaction, or the relation, among multiple systems.
- Nevertheless, relational physics is a complete description of reality.

Rovelli added : This [concept of information] is very weak ; it does not require [consideration of] information storage, thermodynamics, complex systems, meaning, or anything of the sort. In particular :

1. Information can be lost dynamically ([correlated systems can become uncorrelated]) ;
2. [It does] not distinguish between correlation obtained on purpose and accidental correlation ;
3. Most important : any physical system may contain information about another physical system. Information is exchanged via physical interactions. It is always possible to acquire new information about a system. In place of the notion of state, which refers solely to the system, [use] the notion of the information that a system has about another system. Quantum mechanics indicates that the notion of a universal description of the state of the world, shared by all observers, is a concept which is physically untenable, on experimental grounds.

According to the Actor Model, Interaction creates reality. Technical details are in Fig C.1.

The Actor model makes use of two fundamental orders on events [Baker and Hewitt 1977; Clinger 1981, Hewitt 2006]:

1. The *activation order* ($\curvearrowright$) is a fundamental order that models one event activating another (there is energy flow from an event to an event which it activates). The activation order is discrete:

$$\forall e_1, e_2 \in \text{Events} \rightarrow \text{Finite}[\{ e \in \text{Events} \mid e_1 \curvearrowright e \curvearrowright e_2 \}]$$

There are two kinds of events involved in the activation order: reception and transmission. Reception events can activate transmission events and transmission events can activate reception events.

2. The *reception order* of a serialized Actor x ($\rightarrow_x$) models the (total) order of events in which a message is received at x . The reception order of each x is discrete:

$$\forall e_1, e_2 \in \text{ReceptionEvents}_x \rightarrow \text{Finite}[\{ e \in \text{ReceptionEvents}_x \mid e_1 \rightarrow_x e \rightarrow_x e_2 \}]$$

The *combined order* (denoted by $\curvearrowright$) is defined to be the transitive closure of the activation order and the reception orders of all Actors with the following axiom:

$$\forall e_1, e_2 \in \text{Events} \rightarrow \text{Finite}[\{ e \in \text{Events} \mid e_1 \curvearrowright e \curvearrowright e_2 \}]$$

FIGURE C.1 – C. Hewitt answer to T. Bolognesi .

Question from H. Zenil to A. Bauer :

I liked how deeply, in terms of constructive math, you connected the Church-Turing thesis to geometric intuitions about space. I agree that these are strong indicators that the last word on the nature of computation and geometry has not yet been said.

If one of the arguments against constructive mathematics, concerning its expressive power, is not true as you argue in your paper, do you have any hypothesis as to why classical and not intuitionistic mathematics took over and became mainstream? Is it an historical accident, perhaps an inheritance from ancient Greek math? To me, only such an historical accident would explain why mathematicians seem to see it as less exotic to work with infinite arithmetic and oracle machines than with constructive mathematics. If intuitionistic math is richer than classical, how do we reset math in the right direction? Doesn't all this tell us something about the human dimension of mathematics? What does this say about the purported immutable truth of mathematical theorems?

Along the same lines, how historically connected is constructive math to Hilbert's program? In the end Turing's abstract idea of a universal computer, a constructive device that can perform all possible computations, turned out to be technologically realisable. How much more successful can a mathematics be required to be than to embody itself in a physical device capable of carrying out all possible calculations? Would you attribute such an achievement to constructive math and how could a constructive result emerge from such a non-constructive motivation? (the non-constructive negative result of Hilbert's program epitomised by Turing's use of Cantor's diagonal argument). How much can one credit constructive math for being at the foundations of the concept of universal computation, leading to the development of the whole field of computer science? And in that sense what do you make of areas such as computability and algorithmic randomness (or for that matter, modern classical mathematical logic) where most of the human resources are concentrated in highly non-constructive fields?

In this context, do you think the complementary question to the 'Unreasonable Effectiveness of Mathematics in the Natural Sciences' question posed by Eugene Wigner (1960), the question of whether modern math has any effectiveness in the natural sciences, is legitimate today? The usual answer is that one should not ask whether math has any application today, as it might be applicable in the future, and pure research should not be subject to considerations about mundane applications. A legitimate question, however, is to ask how realistic it is to think that ordinal arithmetic or the study of oracle

machines will find any real connection to the natural world in the foreseeable future. Both subjects, however, have given us a lot of insight into math and computation.

Answer by A. Bauer :

Since I am no historian of mathematics I am surely the wrong person to answer authoritatively some of your questions. Nevertheless, I can offer my quasi-informed opinion.

Why did classical and not intuitionistic mathematics take over and became mainstream? This question is a bit like asking why so many people use a particular operating system, or speak a particular language. While many, if not most, mathematicians would like to think that there is something unique and sacred about mathematics, I would advise against such opinions. Mathematics is (still) a deeply human activity. Trends in mathematics are determined mostly by social factors, such as what the popular people do and which kinds of mathematics can get funding.

So yes, it is a historic accident that we use one kind of mathematics rather than another. If engineers invented computers before they invented the steam engine, I am sure mathematics would have developed differently. Your asking how to reset math in the right direction calls for a value judgment which I am not willing to make. A mathematician who is familiar with intuitionistic mathematics will have technical advantages over his purely classical colleagues, but the same may be said for any scientist who knows more than his peers. In any case, if intuitionistic mathematics is going to take over the world of mathematics, it will be partly due to quality of work and mostly thanks to good propaganda and pressure from external sources, such as computer science.

Your next question is about the historical connection between constructive mathematics and Hilbert's program. As far as I know, and I know next to nothing about this particular historical issue, Hilbert's finitism and constructive mathematics were not related historically. You raise an interesting issue when you ask about Turing having developed his machines in the context of classical mathematics. Turing's development of a theory of computation is perfectly constructive, and if there are any non-constructive steps in his papers, they can be easily avoided. Cantor's diagonalization argument, as well as Turing's use of it in relation to the halting problem, are constructively valid (contrary to the popular opinion that any proof which reaches a contradiction is non-constructive). Therefore, it is largely irrelevant whether Turing worked constructively or classically. I will go much further and state that the vast majority of mathematics is independent of foundations, and lu-

ckily so. Algebra and geometry existed before anyone spoke of foundations of mathematics, and I am pretty sure they would not wither away without the 20th century logicians. This of course is not to say that logic and foundations has no role at all. In the 20th century it was a grand unifying force which clarified the nature and methods of mathematics, and gave mathematicians a universal language so that they could speak to each other more easily and discover deep connections that would not have been visible otherwise.

You ask how much one can credit constructive mathematics for being at the foundations of the concept of universal computation leading to the development of the whole field of computer science. Historically speaking, and here again I warn that I am not a historian, constructive mathematics stemmed from a philosophical tradition which was concerned with questions about the nature of mathematical knowledge and how the creative subject, i.e., the mathematician, might recognize mathematical truths. Questions about the nature of computation arose in logic independently, through efforts to clarify the concept of computation. Only later on did Stephen C. Kleene establish a clear connection between intuitionistic arithmetic and computability through his realizability interpretation. In recent times constructive mathematics has had an important and deep influence on computer science, for example through Church's λ -calculus and Martin-Löf's type theory.

Finally, you speak of the unreasonable effectiveness of mathematics in the natural sciences. Is it realistic to think that ordinal arithmetic or the study of oracle machines will find any real connection to the natural world in the foreseeable future? Never say never. Ordinal arithmetic has such a connection already. It is used to show termination properties of programs, for example, and it has undoubtedly inspired the concepts of inductive definition and inductive construction in type theory. And I once heard Robert Soare explain very nicely that an oracle machine is just like a personal computer with access to external information, such as a CD or an Internet connection. Perhaps this is why we keep surfing the Internet, to find an oracle.

But more seriously, you presume that constructive mathematics is unable to deal with ordinal arithmetic and oracles. What evidence is there for such an assumption? In fact, I already mentioned that ordinal arithmetic has its constructive counterpart, namely inductive definitions and constructions, while oracle computations in the context of constructive mathematics happens to be high on my to do list.

Question from H. Zenil to A. Bauer : Klaus Sutner (see his chapter in this volume) seems to provide an example of a constructive proof (that can

in principle be carried out by a Turing machine) offering a solution to Post's problem (for reference : whether there is any r.e. degree strictly between 0 and $0'$), but still unsatisfactory in the terms discussed by Sutner in this volume. For this problem, the novelty of the proof is the ad hoc proof method, but the proof itself says little if anything about the real nature of Post's problem in a more realistic (physical) context, leaving us with what seems to be an unnatural answer that does not provide any insight into the true problem, i.e. whether there are intermediate computational degrees. Sutner concludes that in the context of some reasonable theory of physics, intermediate processes fail to exist. Do you have anything to say in this regard? What bearing does this have on whether constructive math is physically meaningful, or on whether the solution to Post's problem is or isn't truly constructive?

Answer by A. Bauer : I am not aware of any constructive proofs of the existence of intermediate degrees, and I seriously doubt there is one. Sutner does not provide a constructive solution to Post's problem either, and it is not his intent to do so anyhow. In fact, non-existence of a constructive solution would support, not contradict, Sutner's conclusions.

Sutner observes, as others have before, that a certain amount of information hiding is involved in the usual solutions of Post's problem. My reading of Sutner is the following. If hiding of information is prevented, either by the laws of physics or because of other requirements, such as "a 1-dimensional cellular automaton must be used and we can observe its actions", then any computation whose output is an intermediate degree will be recognized as computationally complete when looked at in its entirety by an outside observer. I find this to be a very interesting remark which may go a long way in explaining why intermediate degrees are not observed in nature, provided one looks honestly with eyes wide open.

It is curious to see that information hiding is important both in Sutner's contribution and mine. I argue that a certain amount of information hiding should be provided by the laws of physics, if the Brouwerian continuity and choice principles are to be realized. Similarly, Sutner argues that information hiding could be the root cause of intermediate degrees. Yet, his and my conclusions are in opposition somehow, since for me information hiding prevents the existence of bizarre objects, namely discontinuous maps, while for him it creates them, namely intermediate degrees. The moral of the story seems to be that a proper analysis of real-world computation should seriously take into account the concept of hiding computations and communication channels from external observers.

Question from A. Cabello to S. Wolfram : What if quantum theory turns out to be the “ultimate fundamental theory”, that is, the ultimate “complete set of laws that provide a finite description of how the whole universe works”, you refer to in p. 5 of your contribution? (A Computable Universe)

Answer by S. Wolfram :

Then there’d still be the question of how we perceive definite things to happen in the universe. The formalism of quantum mechanics on its own just gives us only probability amplitudes.

Question from C. Teuscher to S. Wolfram : I’m curious to know what influence the fact that “we will ultimately never be constrained in what we can achieve by the details of our physical universe” have on building artificial brains of some sort?

Answer by S. Wolfram :

It implies that there won’t be any fundamental issue with building artificial brains, whatever those may be. I think it’s a challenge to define what an abstract “artificial brain” is, divorced from all the details of human brains in particular. But I’m confident that in any definition, we’ll be able to build versions even in software.

Question from M. Szudzik to S. Wolfram :

You argue that the evolution of a thermodynamic system from one state to the next is essentially an encryption process. This analogy is compelling because it allows us to use ideas from cryptography to classify models of physical systems. For example, since the evolution is reversible for the classical billiard ball model of a gas, a symmetric-key encryption function can be used as a model for a gas’s evolution. Can you comment further on this analogy?

Answer by S. Wolfram :

I have to say that I viewed encryption as an analogy for the computational processes that lead to the Second Law, not a direct model. Back in the mid-1980s, I realized that the rule 30 cellular automaton could be used as a good source of randomness for a stream cipher. I wondered whether reversible rules could be the basis for a practical encryption system, but I never managed to figure out a way to make that work. Of course, the scrambling by the Second Law is probably a lot weaker than a practical encryption system.

Question from L. De Mol to S. Wolfram :

In his "The computer and the brain", John von Neumann said that :

When we talk about mathematics, we may be discussing a secondary language built upon the primary language truly used by the central nervous system. Thus, the outward forms of our mathematics are not absolutely relevant from the point of view of evaluating what is the mathematical or logical language truly used by the central nervous system.

Do you have any comments on this quote in relation to the idea of reducing physics to mathematics ?

Answer by S. Wolfram :

On this point (if not always on others [see <http://www.stephenwolfram.com/publications/recent/neumann/>]), I think von Neumann is obviously correct. Brains surely don't have direct representations of symbolic integrals, or constructs from predicate logic. Those get built up through, in effect, many layers of software (a concept that arose only after a lot of practical software existed in the world). Whether there are good high-level symbolic ways to represent what's going on in the brain is an interesting question, as is how these relate to human natural language. In my efforts to represent knowledge and what can variously be called "computation" or "thinking", I've been surprised at how simple the representations can be, while still achieving what amounts to human-like performance.

Préface de Roger Penrose à mon recueil "*A Computable Universe : Understanding and Exploring Nature as Computation*" publié par World Scientific, 2012

C'est un grand honneur pour moi d'avoir à présenter la Préface de cette fascinante et merveilleuse collection de commentaires¹, sur la nature de la computation et sa profonde relation avec les opérations de ces lois de base, connues ou inconnues, qui gouvernent l'univers où nous vivons. On lutte en effet ici, de façon fondamentale, avec des questions profondes, et le fait de se retrouver avec tant de points de vue différents est une chose à laquelle on pouvait s'attendre, car en vérité, on en sait peu sur la nature des fondements et sur les origines de ces lois de base, malgré l'immense précision qui se trouve souvent révélée en elles. Par conséquent, il n'est pas étonnant que parmi les points de vue exprimés ici, il y ait de la fière spéculation, bordée de temps en temps par un travail de réflexion partiellement justifié, tandis qu'ailleurs on va trouver une affaire rondement et précisément raisonnée, parfois sous la forme de théorèmes mathématiques rigoureux. Les deux sont comme ils se doivent d'être, car sans un certain travail de réflexion inspirée, difficile de repérer les nouvelles idées vers où se tourner pour vraiment avancer, et sans raisonnement mathématique précis, comme sans observation précise, difficile de savoir quand on a raison-ou quand on a tort, plus souvent. L'année de publication de ce livre, 2012, est particulièrement appropriée, étant le centième anniversaire de la naissance d'Alan Turing, dont les analyses théoriques de l'idée de "machine calculante" ainsi que son travail durant la guerre pour déchiffrer le code nazi, ont eu un énorme impact sur le très grand développement des computers électroniques, et sur l'influence résultante qu'ont eu ces machines sur nos vies et sur la façon dont on se pense nous-mêmes. Cet impact est particulièrement évident avec l'application de la technologie informatique sur les implications des lois physiques connues, que ce soit au niveau des fondements, ou à un plus large niveau comme avec la mécanique des fluides ou la thermodynamique, là où la mise en moyenne de nombres énormes de particules en tant que constituants élémentaires, conduit à des équations dynamiques simples en comparaison. Je dois faire remarquer ici que de temps en temps, on a même suggéré que, dans un certains sens, les "lois" qu'on semble trouver sur la façon dont le monde fonctionne, possèdent toutes cette caractéristique statistique, et que à la racine, il n'existe "pas" de lois physiques sous-jacentes (par ex. la "loi sans loi" de Wheeler [Wheeler, 1983], les idées de "gravité induite" de Sakharov [Sakharov, 1991], etc., et on trouve aussi ce genre de point de vue général dans ce livre²). Cependant, je vois mal ce genre de position avoir beaucoup de chances de fournir quelque chose comme l'extrêmement précise dynamique non-statistique [Schrödinger, 1951]

1. Les notes en pied-de-page citent juste des noms, renvoyant à des chapitres de mon livre (Un Univers Computable par Hector Zenil, Ed.)

2. voir Calude

et la grande sophistication mathématique qu'on trouve si souvent dans la physique du 20ème siècle. Ceci dit, il s'avère que dans des circonstances raisonnablement favorables, les simulations informatiques peuvent mener à des imitations très surprenantes de la réalité, et les représentations visuelles des résultats peuvent à peine se distinguer du réel, un fait dont on se sert souvent au cinéma dans les effets spéciaux, autant que dans les présentations scientifiques sérieuses. Quand on a besoin de précision dans des implications particulières de ce genre d'équations, on peut se retrouver face au difficile problème du comportement chaotique, causant une dépendance exponentiellement de plus en plus sensible aux conditions initiales. Dans ce genre de situations, il y a un aléatoire effectif dans l'évolution du comportement. Pourtant, les simulations computationnelles vont encore donner des résultats pouvant être physiquement permis, et du même coup donner quelque chose de consistant quant au comportement réel. Les simulations informatiques peuvent avoir une grande importance dans de nombreux domaines autres que la physique, comme avec la propagation des épidémies, ou avec l'économie (où les idées mathématiques de la théorie des jeux peuvent jouer un rôle important)³, mais je vais m'occuper ici spécifiquement des systèmes physiques. Le côté impressionnant des simulations informatiques est souvent plus évident quand il s'agit d'une simple mécanique newtonienne du 17ème siècle, dans ses manifestations extrêmement variées. Les implications des lois dynamiques newtoniennes peuvent être largement calculées dans la modélisation des systèmes physiques, même là où on peut se retrouver avec des nombres énormes de particules, tels les atomes d'un gaz simplifié, ou avec des ingrédients de type particulier, telles les étoiles dans les clusters globulaires ou même dans des galaxies entières. On peut remarquer que les simulations informatiques sont normalement faites dans un sens temporel, où le comportement futur se déduit d'une entrée prise dans le passé. En principe, on pourrait aussi réaliser des calculs dans une direction "téléologique" inverse, à cause de la réversibilité temporelle des lois newtoniennes de base⁴. Cependant, à cause du second principe de la thermodynamique, par lequel l'entropie (ou "aléatoire") d'un système physique augmente avec le temps dans le monde réel, de tels calculs à rebrousse-temps ont tendance à être peu dignes de confiance. Quand les lois newtoniennes furent complétées par les équations de Maxwell-Lorentz, pilotant le comportement des champs électromagnétiques et leurs interactions avec des particules matérielles chargées, alors l'étendue des processus physiques pouvant être simulés avec précision par des procédures computationnelles, s'est trouvée grandement augmentée, comme

3. Vellupillai

4. Beavers

avec les phénomènes impliquant le comportement de la lumière visible, ou avec des appareillages touchant aux micro-ondes ou à la propagation radio, ou pour modéliser les vastes nuages de plasma galactique impliquant des flux mélangés d'électrons et de protons dans l'espace, qui peuvent en effet être informatiquement simulés avec une confiance considérable. Ce dernier type de simulation requiert que les équations physiques utilisées s'accordent correctement avec les besoins de la relativité restreinte, en incorporant le point de vue d'Einstein sur la relativité du mouvement et sur le passage du temps. La relativité restreinte d'Einstein a englobé, résumé, remplacé les idées précédentes de Fitzgerald, Lorentz, Poincaré et d'autres, mais même le point de vue d'Einstein a eu besoin d'être reformulé et rendu plus satisfaisant par le changement radical de perspective introduit par Minkowski, qui montre comment les idées de la relativité restreinte s'arrangeaient naturellement dans le cadre géométrique global d'un espace-temps à 4 dimensions. Quand vint à l'esprit d'Einstein l'idée de la relativité générale, où l'espace de géométrie 4 de Minkowski est fondamentalement modifié pour devenir courbe, de façon à pouvoir intégrer les phénomènes gravitationnels, on a trouvé que les simulations des systèmes gravitationnels peuvent se construire avec une précision encore plus grande que ce qu'il était possible de faire avec la théorie newtonienne. La connaissance précise des mouvements planétaires à l'intérieur de notre système solaire est maintenant arrivée à un tel niveau que la théorie du 17ème siècle de Newton ne suffit plus, et on a besoin de la théorie du 20ème siècle d'Einstein. C'est vrai même pour les opérations des Global Positioning Systems, utilisés couramment aujourd'hui, qui seraient inutilisables sans les corrections apportées à la théorie newtonienne par la relativité générale. En effet, peut-être les plus précises des simulations théoriques jamais réalisées, en l'occurrence le pistage du déplacement des étoiles doubles à neutrons, où non seulement on doit prendre en compte les corrections standard de la relativité générale (avance de périhélie, effets Lense-Thirring, etc.) par rapport au mouvement orbital newtonien, mais où on peut aussi calculer, grâce à la théorie, les effets de baisse d'énergie dues aux ondes gravitationnelles (ondulations de la courbure spatio-temporelle), ces simulations se trouvent s'accorder aux mouvements observés avec une précision sans précédent. L'autre révolution majeure en physique théorique fondamentale que le 20ème siècle a amené fut, bien sûr, la mécanique quantique—qu'il faut considérer conjointement avec sa généralisation en théorie quantique des champs, celle-ci requise lorsque les effets de la relativité restreinte et les principes quantiques doivent être pris en compte ensemble. Il est clair d'après beaucoup d'articles de ce livre, que la théorie quantique est considérée (à juste titre) comme d'une importance fondamentale, quand on en vient à l'investigation des opérations sous-jacentes basiques de l'univers physique et de leurs relations avec la computation. Il

existe de nombreuses raisons pour cela, une évidente étant que les processus quantiques sont indubitablement fondamentaux quant aux comportements des plus petits ingrédients de notre univers, et aussi quant aux nombreuses caractéristiques du comportement collectif des systèmes à nombreuses particules, ceux-ci étant spécifiquement de nature mécaniquement quantique, comme le sont l'intrication quantique, la supraconductivité, la condensation de Bose-Einstein, etc. Cependant, il y a une autre caractéristique basique de la mécanique quantique qui peut se compter parmi les raisons de voir cette organisation des choses comme étant plus propice à l'idée de computation que ne l'était la mécanique classique, en l'occurrence la discrétion que la mécanique quantique introduit à la base de la physique théorique. Il semble que dans les premiers jours de la théorie, on fit grand cas de cette discrétion, avec son espoir déduit d'une nature "granulaire" soutenant les opérations du monde physique. On a exprimé l'espoir [[Russell, 1927](#), [Schrödinger, 1951](#)] que, quelque part, la domination de la physique théorique par les concepts de continuité et différentiabilité—qui vont main dans la main avec l'usage envahissant du système à nombres réels—pourrait avoir été au moins brisée, via l'introduction de la mécanique quantique. Par conséquent, on a espéré que les idées de discrétion et de combinatoire pourraient bientôt devenir la force conductrice dominante soutenant les opérations de l'univers, plutôt que la continuité et la différentiabilité dont la physique classique avait été dépendante pendant tant de siècles. Un univers discret est effectivement beaucoup plus en harmonie avec les notions courantes de computation que ne l'est un univers continu, et de nombreux articles de ce livre⁵ défendent effectivement cette perspective, particulièrement dans le contexte des automates cellulaires⁶. Le concept même de "computabilité" apparu au début du 20ème siècle après les travaux de différents logiciens, Gödel, Church, Kleene et beaucoup d'autres, s'appuyant même sur les idées du 19ème de Charles Babbage et Ada Lovelace⁷, et qui fut grandement clarifié par le concept de Turing de machine calculante, et par les idées très proches de Post, ce concept repose en effet sur une discrétion fondamentale des ingrédients de base. Les propositions variées très dépareillées en vue d'un concept effectif de calculabilité, que ces logiciens du début du 20ème siècle ont introduit, se sont toutes avérées équivalentes les unes aux autres, un fait central pour notre vision courante concernant la computation, et que nous donne la thèse de Church-Turing, notamment que cette notion théorique précise de "computabilité" résume effectivement l'idée que nous exprimons instinctivement

5. Bolognesi, Chaitin, Wolfram, Fredkin et Zenil

6. Martinez, Margenstern, Sutner, Wiedermann et Zuse.

7. DeMol, Sieg, Sutner, Swade et Zuse

en parlant de "procédure mécanique" idéalisée. Ce problème est discuté de façon assez profonde par bon nombre d'auteurs dans ce volume⁸. Pour ma part, je me satisfais de la thèse de Church-Turing, au sens originel de cette phrase, notamment que le concept mathématique de computabilité—défini par ce qu'on peut réaliser grâce au λ -calcul de Church, ou de façon équivalente grâce à une machine de Turing—est effectivement l'idéal approprié du concept mathématique dont on a besoin pour nos considérations de computabilité. Que ce soit ou non l'univers dans lequel nous vivons qui fonctionne selon un tel concept de computation, c'est ensuite un problème sur lequel on peut spéculer, ou sur lequel on peut raisonner d'une façon ou d'une autre (voir, par exemple, [Penrose, 1994, Delahaye and Zenil, 2011]). Néanmoins, je peux apprécier qu'il y ait d'autres points de vue là-dessus, et que certains préféreraient définir la "computation" en termes de ce que peut réaliser (en principe?) un objet physique⁹. Pour moi, cependant, la question se pose, et la même question restera certainement posée, quelle que soit notre préférence quant à l'usage du terme "computation". Si on préfère se servir de la définition "physique", alors tout système "calcule" par définition, et dans ce cas il faudrait simplement un mot différent pour le concept mathématique de computation (l'original de Church-Turing), de façon à poser la question en profondeur pour pouvoir étudier la nature peut-être calculable des lois gouvernant les opérations de l'univers. Par conséquent, je vais utiliser ici le terme de "computation" dans ce sens mathématique, et je poserai cette question de la nature computationnelle des lois physiques plus tard, de façon approfondie. Pour en revenir maintenant au problème de la discrétion apparue lors de l'introduction de la mécanique quantique standard, on trouve que cette théorie, telle qu'on la comprend aujourd'hui, ne s'est pas développée dans cette direction fondamentalement discrète, qui se serait si bien ajustée avec nos idées de computation. La discrétion que Max Planck révéla, en 1900, dans son analyse de l'irradiation des corps noirs (bien que non établie initialement de cette manière) était en effet une discrétisation de l'espace des phases—cet espace mathématique hautement dimensionné où chaque degré spatial de liberté, dans un système pluri-particules, s'accompagne d'un degré inertiel (momentum) de liberté correspondant. Ce n'est pas une discrétisation qu'on pourrait directement appliquer à nos perceptions apparemment continues de l'espace et du temps. Néanmoins, différents contributeurs de ce volume¹⁰ se sont aventurés dans cette direction radicale, arguant qu'un certain type de discrétion peut se révéler quand on essaye d'examiner des sé-

8. DeMol, Sieg, Dershowitz, Sutner, Bauer et Cooper

9. Deutsch, Teuscher, Bauer et Cooper

10. Bolognesi et Lloyd

parations spatiales autour de la longueur de Planck l_p (environ 10-35 m) et les séparations temporelles autour du temps de Planck (environ 10-43 sec). Ces séparations sont absurdemment minuscules, plus petites de quelques 20 ordres de magnitude des échelles de distance et temps pertinentes pour les processus de la physique standard des particules. Comme ces échelles de Planck sont énormément en-dessous de ce que tout accélérateur moderne de particules a été capable d'explorer, on peut raisonnablement défendre l'idée d'une granularité dans la structure de l'espace-temps se produisant à des échelles de Planck absurdemment minuscules, et qui n'auraient pas pu se détecter dans les expériences courantes. En plus de cela, certains théoriciens prétendent depuis longtemps, plus particulièrement le distingué et hautement perspicace physicien américain John A. Wheeler [[Wheeler, 1962](#)], que notre compréhension de la façon dont la théorie gravito-quantique doit fonctionner (selon laquelle les principes de la mécanique quantique s'imposent sous le couvert de la théorie générale de la relativité d'Einstein) suppose l'apparition de quelque chose de radicalement nouveau quant à l'espace-temps aux échelles de Planck, et l'image du lisse espace-temps adopté en physique classique devrait être abandonnée pour laisser émerger quelque chose de tout-à-fait différent à ce niveau. L'argument de Wheeler—basé sur des principes venant d'idées conventionnelles sur la façon dont le principe d'incertitude d'Heisenberg s'applique aux champs quantiques—nous pousse à devoir envisager de sauvages "fluctuations quantiques" qui se produiraient à l'échelle de Planck, nous donnant l'image d'un furieux désordre des fluctuations topologiques. Tandis que cette image n'est pas du tout semblable à celle d'un espace-temps discret granulaire, elle soutient au moins l'idée que quelque chose de très différent d'une structure classiquement lisse doit concerner la physique aux échelles de Planck, et il se pourrait qu'une représentation discrète soit vraiment la bonne. C'est un sujet sur lequel il va falloir que je revienne dans cette Préface. Quand il s'agit de simuler des systèmes quantiques traditionnels (sans s'occuper de la nature de la physique à l'échelle de Planck), il se trouve alors, comme c'était le cas avec les systèmes classiques, qu'on a besoin de considérer les solutions continues d'une équation différentielle (partielle)—ici l'équation de Schrödinger. Ainsi, tout comme avec la dynamique classique, on ne peut pas appliquer directement l'idée de Church-Turing sur la calculabilité à l'évolution d'un système quantique, et il semblerait qu'on soit conduit à rechercher des simulations qui soient de simples approximations de l'évolution continue exacte de la fonction d'onde de Schrödinger. Turing lui-même était prudent pour traiter ce genre de problème [[Turing, 1948](#)], que ce soit dans un contexte classique ou quantique, car il pensait en effet que lorsque les approximations discrètes ne sont pas assez bonnes pour telle raison particulière, on pouvait toujours les améliorer tout en restant dans le discret. C'est en effet un des avantages-clé

des représentations numériques par rapport aux analogiques, que de pouvoir réaliser une augmentation exponentielle de la précision d'une simulation digitale simplement en incorporant des digits supplémentaires. Bien sûr, la simulation peut avoir à tourner beaucoup plus longtemps quand plus de digits sont inclus dans l'approximation, mais le problème ici concerne ce qui peut en principe être réalisé par une simulation numérique, plutôt qu'en pratique. Ainsi va l'argument, en théorie, les approximations discrètes peuvent toujours croître en précision, de sorte que les simulations computationnelles d'un processus physique dynamique pourraient être aussi précises qu'on le désire. Personnellement, je ne suis pas pleinement convaincu par ce genre d'argument, surtout quand on simule des systèmes chaotiques. Si on demande simplement à nos simulations de représenter des résultats plausibles, consistants par rapport à toutes les équations physiques associées, en relation avec le comportement du système physique particulier considéré, alors le comportement chaotique peut ne pas être un problème, car on serait seulement intéressé par le caractère réaliste de la simulation, non pas vraiment par la production du résultat détaillé qui va sortir. D'un autre côté, si—comme dans la prévision météo—on insiste pour que la simulation fournisse le vrai résultat du comportement d'un certain système se produisant dans le réel que nous habitons concrètement, alors c'est une autre paire de manches, et les approximations peuvent ne pas suffire, rendant ainsi le comportement chaotique authentiquement problématique¹¹. On peut cependant remarquer que l'équation de Schrödinger, étant linéaire, n'a pas, à strictement parler, de solutions chaotiques. Néanmoins, il existe une notion connue sous le nom de "chaos quantique", qui fait normalement référence à des systèmes quantiques qui sont des quantisations de systèmes chaotiques classiques. Ici, le problème du "chaos quantique" est subtil, et tout se tient dans la question de savoir ce pour quoi on veut utiliser normalement l'équation de Schrödinger, ce qui est en rapport avec l'épineux problème de la mesure quantique. Ce qu'on trouve en pratique, d'une façon générale—et je vais devoir revenir sur ce problème plus tard—c'est que l'évolution de l'équation de Schrödinger ne nous donne pas l'unique résultat qu'on voit apparaître dans le réel, mais nous donne une superposition de résultats alternatifs possibles, avec une valeur de probabilité assignée à chacun. La situation n'est effectivement pas meilleure qu'avec les systèmes chaotiques, et de nouveau nos simulations computationnelles ne peuvent servir à prévoir le résultat dynamique réel d'un système physique particulier. Comme avec les systèmes chaotiques, nos simulations ne vont nous donner que des résultats alternatifs plausibles—dotés de pro-

11. Des développements sur ce problème se trouvent en [Israeli and Goldenfeld, 2004, Wolfram, 2002, Zenil et al., 2012c]

habilités associées—et ne vont normalement pas fournir de prévision claire quant au futur comportement d'un système physique particulier. En fait, la situation quantique est en un sens "pire" qu'avec les systèmes chaotiques classiques, car ici le manque de prédictabilité ne vient pas des limitations de la précision des simulations computationnelles pouvant être menées, mais il se trouve que même une simulation totalement précise de la solution requise de l'équation de Schrödinger ne serait pas capable de prédire en confiance ce que le résultat réel pourrait être. L'unique historique qui émerge, dans l'univers dont avons concrètement l'expérience, n'est qu'un membre de la superposition que l'évolution de l'équation de Schrödinger nous fournit¹². Même cette "simulation précise" reste problématique à certains degrés non négligeables. Il y a encore le problème de l'approximation discrète d'un modèle mathématique d'une réalité fondamentalement continue. Et avec les systèmes quantiques, la simulation précise est confrontée à un problème supplémentaire, en l'occurrence la grande taille de l'espace de paramètres dont on a besoin pour l'équation de Schrödinger d'un système quantique pluri-particules. Cela vient des intrications quantiques citées auparavant. Toutes les intrications possibles entre les différentes particules du système requièrent un paramètre séparé sous forme de nombre complexe, on a donc besoin d'un espace de paramètres exponentiellement vaste, et ça devient rapidement ingérable si on doit retracer tout ce qui se passe. Il se pourrait bien que le futur développement des ordinateurs quantiques trouve ici son application principale, dans la simulation des systèmes quantiques. On trouve dans ce recueil certaines discussions sur le potentiel de ces ordinateurs quantiques, bien qu'il n'en soit sorti encore aucun consensus quant au futur probable de ce domaine intéressant du développement technologique¹³. Nous voyons que malgré la discrétion introduite en physique via la mécanique quantique, nos théories actuelles nous demandent toujours de fonctionner avec des fonctions à nombres réels (ou nombres complexes), plutôt qu'avec des fonctions discrètes. Il existe cependant des propositions (par ex. [Blum, 2012]) où la notion de "computation" est prise dans un sens qui s'applique directement aux opérations à nombres réels, les nombres réels employés en physique théorique étant trai-

12. On peut se demander si l'aléatoire apparent surgissant dans les dynamiques chaotiques classiques ne vient pas d'un véritable aléatoire situé à un niveau quantique plus profond. Cependant, cela ne peut pas clore le chapitre, car l'aléatoire quantique apparaît aussi avec des systèmes classiques quantifiés qui ne sont pas chaotiques. Néanmoins, on peut tout-à-fait spéculer sur le fait que les changements non-linéaires de la mécanique quantique, dont je vais bientôt défendre l'existence, agissent comme un lien entre le comportement chaotique et les aspects probabilistes de la théorie quantique actuelle, et pourraient bien être au cœur de l'affaire.

13. Schmidhuber, Lloyd, Zurek

tés en tant que nombres réels, plutôt que, par exemple, en tant qu'approximations rationnelles de nombres réels (comme des approximations binaires ou décimales terminées de façon finie). De cette manière, on peut mener des simulations de processus physiques sans faire appel aux approximations. Mais on peut être forcé d'avoir des données initiales de la simulation sous la forme de fonctions explicitement connues, et ça peut ne pas être réaliste. De plus, il y a toute une variété de concepts différents de computabilité avec les nombres réels [Blum, 2012, Bridges, 1999, Pour-El and Richards, 1981, Stannett, 2003, Weihrauch, 2000], qui, contrairement à la situation à laquelle on est arrivé avec les variables discrètes (à valeurs entières) où le concept de Church-Turing semble avoir fourni une seule notion de "computation" universelle généralement acceptée, on se retrouve avec plein de propositions différentes pour la computabilité à base de nombres réels, et aucune version unique et acceptée partout de ce genre ne semble évidente. En plus, on s'est malheureusement aperçu que, selon une notion apparemment raisonnable de computabilité pour les nombres réels, l'action de l'opérateur ordinaire d'onde de second ordre s'avérerait incalculable dans certaines circonstances (voir par ex. [Pour-El and Richards, 1981, Pour-El and Richards, 1988]). Quoi que puisse être le verdict final sur la computabilité des nombres réels, il semble qu'on ne se soit pas encore installé dans quelque chose de non-ambigu. Il y a aussi la question de savoir si une théorie exacte de la computabilité à base de nombres réels, aurait une vraie pertinence quant à la façon dont on modélise le monde physique. Car nos mesures de la réalité contiennent toujours une certaine dose d'erreur—que ce soit la limite de la précision d'une mesure ou sur la probabilité qu'un paramètre discret peut prendre selon telle ou telle valeur (comme c'est parfois le cas en mécanique quantique)—je ne vois pas clairement en quoi une telle théorie exacte de la computabilité à base de réels puisse être avantageuse par rapport à nos idées actuelles sur la computation discrète (Church-Turing). Bien que le présent volume n'entre pas dans le débat sur ces sujets, je crois qu'il existe des questions d'importance significative qui ne devraient pas être mises de côté (par exemple, voir [Bridges, 1999, Penrose, 1994, Pour-El and Richards, 1988, Weihrauch, 2000]). Plusieurs articles de ce livre traitent le problème de savoir si, d'une certaine façon, l'univers ne serait pas concrètement un computer¹⁴. Pour moi, cela paraît être une idée assez étrange. Bien que je puisse plus ou moins comprendre ce que signifie le fait que l'univers puisse être computationnellement simulé (théoriquement) pour toutes ses actions physiques¹⁵, ce qui implique une sorte

14. Lloyd, Deutsch, Turner et Zuse

15. Bolognesi et Szudzik

de supposition "constructiviste"¹⁶ pour les opérations du monde physique, je trouve beaucoup moins clair ce que peut signifier pour l'univers d'être un computer. Différentes images viennent à l'esprit, peut-être suggérées par la façon dont on choisit de se représenter le fonctionnement d'un computer électronique moderne. Peut-être que cette représentation est constituée d'un certain nombre de "nœuds" spatialement distincts, reliés entre eux par un système de "fils", le long desquels des signaux voyagent, et avec certaines règles précises opérant au niveau des nœuds, déterminant quelle sortie va apparaître pour chaque entrée possible. On aurait aussi besoin d'avoir un certain type d'accès direct à une aire de stockage efficace non-limitée (ceci étant une partie essentielle des aspirations du modèle de type machine de Turing pour un tel computer). Néanmoins, une telle représentation directe et une telle géométrie stable d'ordinateur n'a pas l'air de ressembler beaucoup aux modèles standard actuels dont nous disposons pour les activités à petite échelle de l'univers où nous nous trouvons. Le côté discret de cette représentation est peut-être un peu plus proche de certains des essais de propositions faits pour un univers physique discret, comme les "ensembles causals"¹⁷ sur lesquels je vais brièvement revenir plus tard, et qui sont des tentatives pour représenter certaines idées radicales de ce à quoi "ressemblerait" l'espace-temps à l'échelle de Planck. Pourtant il existe certaines similitudes entre une telle représentation de type ordinateur et nos théories physiques contemporaines les plus solides. Ces théories nécessitent des constituants individuels, référencés comme "particules quantiques", où chacune aurait un niveau classique de description comme étant spatialement "assimilé à un point"—cependant persistant dans le temps, donnant une représentation classique d'un espace-temps unidimensionnel, du type "monde sur une ligne". Si on conçoit ces mondes linéaires comme jouant le rôle de "fils" dans la représentation ci-dessus s'inspirant d'un ordinateur, alors on peut concevoir des "nœuds" comme les lieux d'interaction (ou points d'intersection) entre les différents mondes linéaires des particules. Cela ne serait pas complètement différent de l'image de computer décrite ci-dessus, bien que dans la théorie standard, la géométrie topologique des connections pour les nœuds et les fils fasse partie de la dynamique, et ne soit pas fixée à l'avance. Peut-être que le manque de géométrie fixe des connections donnerait une représentation ressemblant plutôt à un type amorphe de structure d'ordinateur considéré aussi dans ce livre¹⁸, plutôt qu'à un ordinateur conventionnel. Cependant, il n'est toujours pas évident de comment représenter "l'accès direct à une zone

16. Bauer

17. Bolognesi

18. Hewitt, Jiri, Teuscher, Margenstern et Wiedermann

de stockage potentiellement illimitée". Plus sérieusement, c'est simplement la représentation classique qui est évoquée dans nos descriptions de l'activité des particules à cette échelle, où " l'image " quantique consisterait (plus ou moins) en une superposition de toutes ces représentations classiques, chacune pondérée par un nombre complexe. Une telle "image" arrive peut-être à être un peu plus proche de la façon dont un ordinateur quantique¹⁹ peut se représenter, mais encore une fois il nous reste des problèmes cruciaux soulevés par la topologie des connexions comme faisant partie de la dynamique, et l'absence d'une "aire de stockage illimitée", dans la représentation physique, ce qui me semble constituer des différences fondamentales entre notre représentation de l'univers et un ordinateur quantique. Ajouté à tout ceci, il y a encore le problème de comment traiter le continuum de façon computationnelle, qui dans la théorie quantique (des champs) est plus à proprement parler un complexe plutôt qu'un continuum concret. Et par dessus tout, on a le problème de comment récupérer l'information d'un système quantique. Cela demande une analyse du problème de la mesure sur lequel je vais avoir à revenir bientôt. Je pense que, malgré tout, quand les gens font référence à l'univers comme "étant" un ordinateur, l'image qu'ils ont à l'esprit n'est pas aussi précise que ce que je viens de suggérer. Plus vraisemblablement, pour cet "univers-ordinateur", il se peut simplement qu'ils pensent que d'une part, les actions de l'univers peuvent être précisément simulées à tous égards, mais que d'autre part, l'univers n'a pas d'autre qualité fonctionnelle, distincte de ce comportement computationnel. Plus en détails, pour notre "univers-ordinateur", ce serait comme s'il y avait un certain paramètre t (surement discret, pouvant prendre des valeurs entières) qui décrirait le passage du temps (une idée pas très relativiste!), et l'état de l'univers à un moment quelconque (par ex. à une t -valeur) aurait une certaine description computationnelle, qui pourrait s'encoder complètement grâce à un seul nombre entier naturel St . Ce serait le boulot de l'univers que de calculer St' à partir de St , à chaque fois que $t' > t$, et le fait que l'univers soit considéré comme étant un computer impliquerait non seulement qu'il soit toujours capable de réaliser ce passage, mais—plus important—impliquerait que ce soit la seule fonction de l'univers. D'un autre côté, il me semble à moi que si l'univers possédait une quelconque fonction supplémentaire, telle qu'assigner une réalité à tout aspect de cette description, alors il ne serait pas simplement un computer, mais serait quelque chose de plus, en réussissant à nous donner une sorte d'ontologie allant au-delà de la simple description computationnelle. Pour conclure cette Préface, je désire présenter quelque chose qui est plus en ligne avec mes propres vues au sujet de la relation entre computation et nature de la réalité

19. Schmidhuber, Margenstern, Zurewsky

physique. Pour commencer, je devrais peut-être souligner que mes points de vue ont considérablement évolué, mais sans trop de changements abrupts. Assez tôt, j'ai été plutôt fermement persuadé qu'il devait y avoir une base discrète ou combinatoire derrière la physique, peut-être quelque chose qui corresponde à certains des articles¹ de ce livre. En 1967, Erwin Kronheimer et moi avons publié un papier [[Kronheimer and Penrose, 1967](#), [Penrose, 1967](#)] sur le type d'ensembles causals dont il est fait référence plus tôt dans cette Préface, où les relations basiques entre les éléments sont celles de la causalité [[Arkani-Hamed et al., 2010](#)], reflétant les relations causales entre les événements dans l'espace-temps continu, mais où n'est supposée aucune continuité ou régularité, et où on pouvait même envisager des situations dans lesquelles le nombre total de ces éléments soit fini. Bien que j'ai aussi d'autres raisons de m'intéresser à des espaces dotés d'une structure définie seulement par des relations causales—en partie en vue de leur rôle en rapport avec les théorèmes de la singularité [[Hawking and Penrose, 1970](#), [Penrose, 1965](#)](pour l'étude des trous noirs et de la cosmologie)—les relations de causalité n'étant pas nécessairement liées à l'idée d'une organisation lisse de l'espace-temps, je ne m'attendais pas trop à ce que la véritable structure à petite échelle de notre univers concret soit décrite avec bonheur en ces termes. J'avais pensé qu'il était plus probable qu'une idée combinatoire différente, que j'avais bien appris à manipuler un peu avant, en l'occurrence l'idée que des réseaux de spins (voir [[Penrose, 1971](#)]) pouvaient parfaitement fonctionner comme base pour la physique (et d'ailleurs, beaucoup plus tard, une version de la théorie des réseaux de spins a fait partie de l'approche dite à variables de boucle pour modéliser la gravité quantique [[Ashtekar and Lewandowski, 2004](#)], bien que le rôle endossé par les réseaux de spins dans le cadre de la gravité à variables de boucle soit un peu différent de ce que j'avais envisagé au départ). La théorie du réseau de spins fut basée sur l'un des aspects les plus frappants de la mécanique quantique standard, c'est une notion fondamentale qui est de l'ordre du continu en mécanique classique, et de l'ordre du discret en mécanique quantique, en l'occurrence le moment cinétique (ou spin). En fait, nombreuses parmi les plus basiques et contre-intuitives caractéristiques de la mécanique quantique, comme la discrétion et la non-localité (de Bell)²⁰, sont exprimées plus puissamment en termes de spins quantiques. La curieuse relation entre l'étalement continu des possibilités que peut prendre l'orientation d'un axe de spin dans notre imagerie classique de l'espace-temps, et la nature discrète ("granulaire") de l'idée quantique "d'orientation d'axe de spin", m'a toujours fasciné pour de bon. Ça et la non-localité de l'information de la mécanique quantique sont arrivés ensemble dans la théorie des

20. Breuer, Cabello, Schmidhuber et Zenil

réseaux de spins, dans laquelle l'idée classique (continue) d'une "orientation" spatiale n'apparaît pas de façon bien définie jusqu'à ce que de très grandes structures de réseaux de spins soient présentes, pour fournir une bonne approximation de la sphère continue des orientations spatiales possibles. Des montages mathématiques spécifiques pour calculer des expressions souvent extrêmement compliquées, on été développés, mais tout reste complètement discret, et computationnel au sens conventionnel du terme, la continuité n'apparaissant qu'à la limite des grands nombres. Le besoin de généraliser l'idée des réseaux de spins pour que la géométrie à 4 dimensions de l'espace-temps puisse être décrite, plutôt que juste la sphère des orientations spatiales, a fini par trouver un peu satisfaction dans les idées de la théorie des twisteurs (voir [Penrose, 1967, Penrose, 2014], chapitre 33). Cela a donné une manière différente de voir la géométrie de l'espace-temps, de ce qu'on avait l'habitude—mais maintenant l'idée de discrétion sous-tendant les bases de la physique commence à décliner, pour être supplantée par la magie de la géométrie complexe et de l'analyse. On conçoit normalement l'espace-temps comme étant composé "d'événements" (par ex. de points spatio-temporels), qui sont les éléments de base de la géométrie. A la place, la théorie des twisteurs conçoit ses éléments de base comme modélisés sur les historiques complets de particules à spin, sans masse et en vol libre. D'après une combinaison soigneuse, mélangeant les idées de la géométrie-4 de l'espace avec celles de la structure quantique des moments cinétiques relativistes de particules sans masse, on a développé [Penrose, 2005, Penrose, 1971] le concept "d'algèbre des twisteurs". En relativité restreinte, le concept de twisteur, qui décrit la structure cinématique d'une particule à spin et sans masse, trouve sa description mathématique comme élément de l'espace complexe T à 4 vecteurs, référencé comme un "espace des twisteurs". La géométrie de T est reliée à la géométrie réelle de l'espace-4 M de Minkowski, au moyen d'une correspondance géométrique explicite, faisant la liaison directe entre M et la géométrie-3 complexe de l'espace twisteur projectif PT . Il se trouve que les nombres complexes de la mécanique quantique coïncident de façon surprenante avec ceux de la géométrie complexe de la théorie des twisteurs, et il y a une intrigante interrelation entre la non-localité qui apparaît naturellement dans la description par twisteurs des fonctions d'onde quantiques et la non-localité qu'on rencontre concrètement dans les phénomènes quantiques [Penrose, 2005]. Ces dernières années, la théorie des twisteurs s'est révélée d'une grande valeur pour le calcul des processus de dispersion à haute énergie, où les masses au repos des particules impliquées peuvent être ignorées, (voir par exemple [Arkani-Hamed et al., 2010]), mais nombreux sont les problèmes plus importants auxquels la théorie des twisteurs est confrontée, qui restent non-résolus. Ça a toujours été un but de la théorie des twisteurs

(seulement partiellement atteint) que de pouvoir donner forme à un véhicule capable d'unir naturellement la mécanique quantique avec la relativité générale. A côté de ça, je n'emploie pas "gravité quantique" au sens conventionnel selon lequel ce terme est utilisé. Ce qu'on entend habituellement par gravité quantique, c'est une sorte d'assemblage dans lequel les idées de la théorie d'Einstein sur la gravité—nommément la relativité générale (ou alors peut-être certaines versions modifiées de la théorie d'Einstein)—viennent se mettre sous la protection de la théorie quantique des champs. Ce point de vue consiste à considérer les lois de la théorie quantique des champs comme inviolées, et à dire que les concepts de la relativité générale doivent produire ceux de la théorie quantique via une certaine forme appropriée de "quantisation". Ma vision personnelle a toujours été différente de cela, parce que je pense que la théorie quantique elle-même, assez indépendamment de la nécessité pour elle d'être unifiée avec la théorie de la relativité générale, est auto-inconsistante à la base et qu'elle a besoin d'une certaine aide se trouvant à l'extérieur des règles normales de la théorie quantique (des champs). La vision défendue ici est que des principes sous-jacents de la relativité générale devraient contribuer à apporter cette assistance extérieure. Cette inconsistance est des plus fondamentales, et elle est complètement évidente au premier sens du terme (un "éléphant dans le salon"!) comme nous allons le voir. Tel que je l'ai fait remarquer plus tôt, on considère l'évolution d'un système quantique isolé comme étant gouverné par l'équation de Schrödinger—ou évolution unitaire, pour se servir d'un terme plus général—pour laquelle j'utilise le symbole "U". Mais, comme remarqué plus tôt, la réalité du monde tel qu'observé autour de nous a tendance à ne pas se décrire directement par la solution ψ de cette équation, obtenue par cette U-évolution, mais quand une observation ou une "mesure" est supposée avoir été faite, on considère que ψ a "sauté" sur un seul membre ψ_r d'une famille de solutions alternatives superposées

$$\psi = \alpha_1\psi_1 + \alpha_2\psi_2 + \dots + \alpha_n\psi_n$$

où les modules au carré respectifs des pondérations à nombres complexes $\alpha_1, \alpha_2, \dots, \alpha_n$, donnent les probabilités respectives de chaque ψ_r en tant que résultat (les quantités ψ_r étant supposées être toutes normalisées et mutuellement orthogonales). Le "processus d'évolution" par lequel ψ est remplacé par un ψ_r particulier qui se trouve émerger, est la réduction de l'état (l'effondrement de la fonction d'onde) et je dénote ce processus²¹ par la lettre "R". Bien

21. Dans l'ouvrage classique de Von Neumann, *Fondements Mathématiques de la Mécanique Quantique* [Von Neumann, 2013], l'auteur introduit "R" et "U" sous les dénominations respectives de "processus I" et "processus II".

sûr, il va y avoir beaucoup de ce genre de décompositions, pour un ψ donné, et cela dépend du choix de la base qui est supposée être déterminée par le choix de "l'appareillage de mesure". En effet, on doit accepter le fait que l'appareil de mesure fasse partie du système complet en considération, et donc il devrait y avoir un état quantique devenant intriqué avec le système quantique en examen. Néanmoins, on considère toujours qu'il y a un "saut" dans le système dans son entier, dès que la mesure est considérée comme ayant été faite, où les différents "états des pointeurs" de l'appareil sont intriqués avec les différents ψ_r possibles pouvant apparaître comme résultats. Il est évident que ce "saut" de l'état du système (constitué à la fois par l'appareil de mesure et par le système en examen, eux-mêmes plongés dans tout un environnement en rapport avec eux), apparaissant dans un moment borné par avant-la-mesure et après-la-mesure, n'est pas continu, normalement, et laisse venir une seule des solutions de l'équation de Schrödinger : donc R viole U de façon flagrante (et ce dans presque toutes les circonstances). Pourquoi les physiciens ne considèrent-ils pas normalement cela comme une contradiction interne de la mécanique quantique ? Les réponses sont nombreuses, impliquant habituellement un certain problème subtil "d'interprétation", à partir duquel les physiciens essayent de contourner cette (apparente ?) contradiction. C'est là où le point de vue "multi-univers" de Hugh Everett III est souvent invoqué [DeWitt et al., 1973, Everett III, 1957], grâce auquel on considère que tous les résultats alternatifs coexistent simplement(!) dans une superposition quantique, et qu'on touche peut-être là à une caractéristique de nos processus de perception consciente faisant qu'on ne perçoit toujours qu'une de ces alternatives. Malgré le succès de cette idée auprès des physiciens ayant une tournure d'esprit philosophique (ou auprès des philosophes calés en physique), je trouve ce point de vue très insatisfaisant. J'admets volontiers que c'est effectivement là où on arrive, si on prend les U-processus comme inviolables, mais pour moi c'est faire une *reductio ad absurdum* et ça indique clairement le besoin de chercher à améliorer la mécanique quantique courante. Pour présenter les choses autrement, même si le point de vue des multi-univers est "correct" dans un certain sens, il reste inadéquat en tant que description du monde physique, pour la simple raison qu'en l'état, il ne décrit pas du tout le monde tel qu'on l'observe, dans lequel d'ailleurs on trouve quelque chose qui est bien évalué par les processus R et qui se produit concrètement quand on a affaire à des superpositions quantiques d'états suffisamment différentes les unes des autres.

Qu'est-ce que j'entends par "suffisamment différents" ? Il est clair qu'une simple différence de distance physique, correspondant aux différents déplacements matériels impliqués dans les états superposés, n'est pas le bon

critère, car il y a eu des expériences confirmées dans lesquelles les états de photons distants de dizaines de kilomètres maintenaient encore leurs intrications quantiques l'un avec l'autre, donc que leurs différents états polarisés possibles restaient en superposition quantique même à ces distances [Ratcliff et al., 2012]. Toutefois, il y a des raisons de s'attendre à ce que, d'après divers principes de base de la relativité générale d'Einstein (voir [Diosi, 1989, Penrose, 1996, Penrose, 2009]), lorsque les déplacements de masse entre deux états quantiques superposés deviennent grands, on ait alors l'apparition d'une instabilité de ces superpositions, qui finissent par se désintégrer en un temps τ calculable, choisissant l'un ou l'autre état, de sorte que le comportement classique commence à prendre le pas sur le comportement quantique. L'estimation de τ est donnée par la formule

$$\tau \approx h/E_G$$

où E_G est l'auto-énergie de la différence entre les distributions de masse dans chacun des deux états quantiques en jeu, chacun supposé constituer un état stationnaire en propre. Une telle désintégration représenterait une déviation de la linéarité standard de U et pourrait bien être le résultat d'une espèce de comportement chaotique apparaissant dans une certaine généralisation non-linéaire de la mécanique quantique actuelle. Il existe des expériences en cours dédiées pour tester cette proposition, et nous attendons des résultats dans les prochaines années (voir [Marshall et al., 2003]). Pour diverses raisons, touchant en partie à la non-localité quantique dont il est fait référence plus tôt (qui commence seulement à poser des problèmes substantiels pour le réalisme quantique lorsque R est impliqué, traité comme un phénomène réel), je m'attends à ce que ce changement dans la mécanique quantique courante représente une révolution majeure et qu'il ne soit pas du tout facile d'y arriver en simplement "bricolant" les équations de Schrödinger. En effet, je m'attends à ce qu'une telle théorie soit non-calculable d'une certaine façon très subtile. Pourquoi m'engager dans une telle affirmation ? Les raisons principales sont plutôt compliquées, et je comprends fort bien pourquoi certaines personnes voient mes propositions comme quelque peu bizarres. Néanmoins, j'ai la vision qu'il existe une bonne raison rationnelle et fondamentale de croire que cette direction pourrait être la bonne ! A la base, la raison vient des fameux théorèmes d'incomplétude de Gödel, que je considère comme fournissant un exemple puissant pour la compréhension humaine de quelque chose d'essentiellement non-calculable. L'argument central est un argument familier, et j'ai toujours du mal à comprendre pourquoi tant de gens ne veulent pas accepter ce qui semble être sa conséquence très claire.

En termes simples, l'argument peut s'appliquer à nos capacités à démontrer vraies certaines propositions mathématiques—qui peuvent prendre la forme de Π_1 -phrases. Une Π_1 -phrase est une affirmation que certaines computations proposées par Turing ne se terminent jamais (des exemples étant le "dernier théorème de Fermat" de Wiles et le théorème de Lagrange disant que tout nombre naturel est la somme de quatre carrés). On peut essayer de résumer, à l'intérieur d'une certaine procédure algorithmique A , tous les types possibles d'arguments pouvant, en principe, être utilisés pour établir les Π_1 -phrases, en accord avec la perspicacité et la compréhension humaine. Cet argument serait une preuve dans un certain système formel donné F , où A serait un algorithme pour vérifier si une preuve proposée utilisant les règles de F a bien été testée correctement, en donnant la réponse OUI après un nombre fini d'étapes, si c'est effectivement le cas. Ce que le théorème de Gödel(-Turing) montre, dans ce contexte, c'est que si on a confiance en A (et donc confiance en la solidité d'un F , par apport aux Π_1 -phrases), et qu'une Π_1 -phrase "prouvée" est en effet vraie chaque fois que A affirme OUI, alors on peut explicitement admettre une Π_1 -phrase G où notre confiance en A s'étend aussi à la confiance en la vérité de G , même si A lui-même se trouve incapable d'établir directement G . Dans le cas du second théorème d'incomplétude de Gödel, la Π_1 -phrase $G(=G(F))$ serait une affirmation de la consistance de F , et G serait l'affirmation que, parmi les théorèmes de F , il n'y en a aucun dont la négation serait aussi un théorème de F . Bien que notre confiance nous dise que A serait incapable d'établir G (c-à-d la consistance de F), notre confiance que G est réellement vraie vient de notre confiance en A (qui dépend de la consistance de F —autrement F serait capable de poser $2 = 3$, une conclusion en laquelle nous n'aurions certainement pas confiance). Notre confiance dans l'utilisation de F comme moyen d'établir la vérité de Π_1 -phrases nous transporte donc au-delà des capacités directes de F , et nous permet d'affirmer que $G(F)$ est vraie, sur la base de cette confiance, malgré le fait que F ne contienne pas $G(F)$ parmi ses théorèmes. C'est en gros l'idée principale de l'attaque de Gödel sur le formalisme. Bien que la formulation des divers domaines des mathématiques ait certainement une valeur, en nous permettant de transférer différents aspects de la compréhension humaine et de la sagacité à l'intérieur de procédures calculables, Gödel nous montre que ces procédures explicites, une fois connues—et de confiance—ne peuvent pas couvrir tout ce qui est accessible en mathématique à notre compréhension et notre perspicacité. Et effectivement, cela s'applique déjà au domaine relativement limité des Π_1 -phrases. Pourtant cela ne suffit pas à démontrer que la pensée humaine est, à la racine, une procédure non-algorithmique, et je vais lister ici les arguments principaux contre cette idée de démonstration, c-à-d les critiques contre la déclaration que les arguments du type Gödel montrent

une compréhension humaine non-computationnelle.

1. Argument des erreurs—les mathématiciens font des erreurs, donc les arguments rigoureux de type Gödel ne s'appliquent pas.
2. Argument de l'extrême complication—les algorithmes gouvernant la compréhension mathématique humaine sont si immensément compliqués que les déclarations de Gödel sont complètement en deçà.
3. Argument de l'ignorance de l'algorithme—nous ne connaissons pas le processus algorithmique qui sous-tend notre compréhension mathématique, donc nous ne pouvons pas construire sa Gödel-déclaration.

J'ai essayé ailleurs [Penrose, 1994] de défendre que (1), (2) et (3) ne suffisaient pas à valider la conclusion que notre compréhension consciente est entièrement le produit d'actions computationnelles, et ce n'est pas mon but d'en répéter ici le détail. Néanmoins, je vais brièvement résumer mes contre-arguments dans ce qui suit. Le point principal par rapport à (1), est que les erreurs humaines sont corrigeables. On n'a pas de gros problèmes avec les tâtonnements souvent erronés dont les mathématiciens se servent dans leur quête de vérité, mais plus problématiques sont les idéaux auxquels surtout ils mesurent leurs résultats. C'est leur capacité à percevoir ces idéaux qui pose problème, si ce n'est en principe, et c'est cette capacité à percevoir une vérité mathématique idéale qui est en jeu ici, pas les erreurs que nous faisons tous de temps en temps. (Il peut sembler d'après mon commentaire, que je vois vraiment la vérité mathématique—plus particulièrement sur des matériaux aussi directs que les Π_1 -phrases—comme quelque chose d'absolu, et d'extérieur à nous-mêmes. Mais j'apprécie que d'autres soient parfois moins sympathiques avec ce type de point de vue. Cependant, je ne crois pas qu'une position philosophique par rapport à ça puisse affecter significativement les arguments que j'avance.) Par rapport à (2), cela revient à peu près au même. Si les algorithmes sont connus en principe, alors leur taille ou complication ne pose pas de réel problème. Cela s'applique à un grand nombre d'arguments mathématiques. Dans la preuve d'Euclide sur l'infinité des nombres premiers, par exemple, on doit considérer des nombres premiers qui sont si grands qu'il n'y aurait pas moyen de seulement les écrire entièrement à l'intérieur de tout l'univers observable, et d'en calculer le produit est d'autant plus hors de question. Mais ceci n'a rien à voir avec la preuve. La même chose s'applique à (2). L'argument (3) est cependant beaucoup plus dans le sujet, et fut à l'origine une réserve émise par Gödel lui-même (voir les commentaires cités là) quant à en tirer la forte conclusion que je défends ici. Plutôt que de poursuivre plus loin l'argument logique, ce qu'on peut certainement faire (voir [Penrose, 1994]), je dois simplement

indiquer l'extraordinaire improbabilité de l'action algorithmique nécessaire devant émerger dans nos têtes, par le processus de sélection naturelle. Un tel algorithme devrait être d'une sophistication extraordinaire, pour être capable de contenir dans son "système formel" effectif de nombreuses étapes de "gödelisation". Par exemple, j'ai souligné ailleurs [Penrose, 1998] que, bien que le théorème de Goodstein [Goodstein, 1944], dont le sens est aisément accessible même à ceux dotés de peu de connaissances en mathématiques autres que la notation numérique de base, ait été démontré par Kirby et Paris [Kirby and Paris, 1982] comme inaccessible à l'arithmétique de premier ordre de Peano (sans une étape de "gödelisation", je précise), ce théorème peut quand même être considéré comme vrai selon toute compréhension mathématique. Si notre compréhension mathématique se produit par le biais d'un certain processus algorithmique (inconnu mais judicieux), son apparition par sélection naturelle reste un mystère total, lors que les expériences de nos lointains ancêtres n'auraient tiré aucun bénéfice d'avoir eu planté dans leurs cerveaux un algorithme à la fois si sophistiqué et si peu les concernant ! Ou alors si on accepte que notre compréhension mathématique n'est pas un processus algorithmique, on doit se poser la question de savoir de quel type de processus il s'agit. Un point-clé, me semble-t-il, c'est qu'une authentique compréhension (au moins au sens courant de ce mot) est quelque chose qui relève de la conscience—et l'emploi du mot "compréhension" me semblerait être un abus de langage s'il fallait l'appliquer à une entité qui n'aurait pas une réelle conscience du sujet qu'elle est en train de traiter. Ce que j'appelle conscience [awareness] est la conscience primordiale et perceptive, l'attention, la forme passive de la conscience réflexive [consciousness], donc il me semble que c'est le développement évolutif menant à la conscience réflexive qui est la clé, et qu'une telle qualité peut certainement avoir émergé par sélection naturelle, en étant capable de donner un énorme avantage sélectif aux créatures la possédant. Disant cela, j'exprime la vision que la conscience réflexive est effectivement fonctionnelle et n'est pas un "épiphénomène" apparu simplement pour accompagner certains types de processus cognitifs. Cette vision est assurément une conséquence du niveau de "compréhension" nécessitant de l'attention consciente, car la compréhension est certainement fonctionnelle. Qu'il soit clair que je ne prétends pas connaître—ou être capable de définir—ce qu'est la conscience, mais son rôle dans la "compréhension" sous-jacente (quelle qu'elle soit) me semble être d'une grande importance dans l'évolution, et elle pourrait bien avoir émergé comme produit de la sélection naturelle. Je dois aussi préciser que je m'intéresse à la question de la conscience en tant que question scientifique, et que je n'adopte pas le point de vue considérant cette matière comme étant inaccessible à l'investigation scientifique. Je pose aussi que des cerveaux hu-

mains sains et éveillés (ainsi que n'importe quel autre type de cerveau animal s'avérant aussi en être capable) peuvent, d'une façon ou d'une autre, susciter la conscience par l'application de ces mêmes lois physiques à l'œuvre dans l'univers, même si la conscience elle-même survient seulement dans des circonstances très spéciales d'organisation nécessaire pour promouvoir son apparition. De quel type de circonstance s'agit-il, qui favoriserait l'émergence d'un certain type d'action non-calculable—si on garde à l'esprit que les équations différentielles déterministes de la physique classique ou quantique semblent être essentiellement de nature calculable? Ma réponse à cette question est que la non-calculabilité doit se tenir dans des lois pas encore découvertes, et qui pourraient être concernées ici. (Je connais mal le problème, cité plus haut, de la simulation computationnelle discrète d'une évolution continue. Néanmoins, j'accepte l'idée qu'il puisse y avoir certains points de grande pertinence devant être plus approfondis.) Autant que je sache, la seule grande inconnue, dans les lois physiques, qui pourrait relever d'une authentique pertinence ici, est l'énigme U/R de la mécanique quantique, citée ci-dessus. Dans presque tous les processus qui s'y tiennent, on n'a pas besoin qu'une présumée Nouvelle Théorie aille au-delà de la mécanique quantique courante, surtout parce que ses effets seraient alors indétectables, étant noyés par les influences aléatoires très variées de la décohérence environnementale. Mais dans le cerveau, il pourrait y avoir des structures spéciales capables de préserver la cohérence quantique sur une durée qui rendrait le critère mentionné auparavant $\tau \approx h/E_G$ tout-à-fait pertinent. Dans ce cas, l'action normale purement probabiliste que les processus R de la théorie quantique standard nous fournissent, serait à remplacer par une certaine subtile décision non-calculable dont le choix conduirait à la réduction d'état. Avec une organisation cérébrale, où les réponses synaptiques sont sensibles à ces choix, on peut imaginer que les réponses du cerveau soient en effet utilement non-calculables. Ceci est le fondement du schéma de "Réduction Objective Orchestrée" (RO-Orch) que Stuart Hameroff et moi avons proposé il y a quelques années, où les "structures spéciales" seraient des microtubules neuronales de type approprié (voir [Hameroff and Penrose, 1996, Penrose, 1996, Penrose and Hameroff, 2011]). Il est à peine surprenant qu'une telle proposition ait été accueillie avec grand scepticisme, pour la très simple raison que pour avoir de la cohérence quantique à température du corps et à peu près au niveau requis, il faut aller bien au-delà des attentes des calculs physiques standard appliqués aux modèles cellulaires simplifiés [Tegmark, 2000]. Cependant les cellules biologiques sont, en fait, des structures hautement sophistiquées, et on peut raisonnablement s'attendre à ce que, quand les structures de certaines parties de la cellule sont spécialisées dans le bon sens, leur comportement puisse montrer des propriétés assez inhabituelles de type

quantique . En fait, des expériences récentes menées au Japon par Anirban Bandyopadhyay [[Sahu et al.](#),] et ses collègues semblent avoir démontré que de fascinants effets de cohérence quantique se produisent concrètement à température ambiante dans les microtubules des neurones. Pour l'instant, ce sont des résultats préliminaires, mais ils paraissent encourager le schéma de la RO-Orch, et il va être très intéressant de voir comment les choses vont évoluer. Même si tout ceci vient à être accepté, il faudra quand même se demander à quoi peuvent bien servir ces petites actions non-calculables, sporadiques, vis-à-vis du fonctionnement cérébral. Elles auraient en effet peu de valeur, à moins que la cohérence quantique soit une caractéristique très générale, impliquant de grandes zones du cerveau, et le processus devrait opérer d'une façon globalement cohérente. C'est d'ailleurs la représentation des RO-Orchs, et nous pensons que les moments de conscience se produisent lorsqu'il y a réduction d'état sur de nombreux sites (dans les microtubules) simultanément et d'une manière orchestrée, de sorte que les potentiels synaptiques soient influencés en de nombreux endroits et qu'il en résulte une influence concertée, ce à quoi on s'attendrait pour des actions conscientes. Les résultats d'actes particuliers de compréhension consciente ne seraient probablement pas quelque chose de simple, et dépendraient de l'expérience des mémoires ainsi que de la logique. Mais l'ingrédient non-calculable est supposé être essentiel, pour les raisons décrites ci-dessus. Selon ce point de vue, nos actions conscientes font partie de la physique—contenues dans une Nouvelle Théorie inconnue aujourd'hui dans le détail. L'impact de cette théorie sur les processus non-organisés de cette façon n'est pas évident. Cela devrait laisser des traces sur les systèmes—comme les cerveaux humains sains et éveillés—et faire émerger actions et perceptions conscientes. Les effets non-calculables de cette Nouvelle Théorie apparaîtraient ainsi et donneraient des actions décrites comme "hypercalculées". Ces actions hypercalculées seraient-elles éloignées du schéma normal de la physique calculable ? Comme l'éclair Gödélien, nous permettant de transcender un système formel F donné de confiance, nous fournit cet éclairage sous la forme d'une Π_1 -phrase, nommément $G(F)$, on pourrait s'attendre à ce que nous puissions modéliser de telles actions hypercomputationnelles sous la forme d'une machine-oracle de Turing, où l'oracle est capable d'affirmer la vérité ou la fausseté d'une Π_1 -phrase. Toutefois cela ne serait pas suffisant (ni, semble-t-il, nécessaire), car on peut encore appliquer un éclairage de "diagonalisation" de type Gödel sur les machines-oracle à Π_1 -phrases pour les transcender elles aussi. Dans un article récent [[Penrose, 2012](#)], je considère un type d'oracle que je qualifie d'"oracle prudent", dans le but de modéliser de manière un peu plus précise le genre de choses que pourrait imaginer des mathématiciens humains, où l'oracle prudent peut examiner une Π_1 -phrase (au sujet de n'importe quel

nombre naturel n) et répondre qu'elle est soit "vraie", soit "fausse" (avec une nécessaire confiance dans chaque cas), ou bien confesser être incapable de fournir une réponse, ou encore échouant partout, continuer simplement à réfléchir indéfiniment sans jamais donner de réponse du tout. Encore une fois, une diagonalisation de type Gödel permet à notre sagacité de transcender les capacités de n'importe quel montage de ce genre ! Quelles que soient les types de capacités hypercalculatoires conférées à une telle "Nouvelle Théorie", il semble que ce soit quelque chose de très subtil. C'est une sorte de capacité sans terminaison donnant la possibilité de "reculer" et d'observer n'importe quelle structure ayant été auparavant considérée. Cela semble être une qualité que la conscience possède, mais comment incorporer ce genre de chose dans une théorie physique est difficile à imaginer, dans l'état de nos théories actuelles.

Annexe D

Calculating Space (*Rechnender Raum*)¹

1st. re-edition² written in L^AT_EX by A. German and H. Zenil

2. with kind permission by all parties involved, including MIT and Zuse's family.

DEDICATION TO DR. SCHUFF

The work which follows stands somewhat outside the presently accepted method of approach, and it was for this reason rather difficult to find a publisher ready to undertake publication of such a work. For this reason I am indebted to the Vieweg Press and especially to Dr. Schuff for undertaking publication. Dr. Schuff suggested that a summary be printed in the Journal “Elektronische Datenverarbeitung” (Electronic Data Processing), which appeared last year.

The tragic death of Dr. Schuff has deeply shaken his friends, and we will always remember him with affection.

i Introduction

It is obvious to us today that numerical calculations can be successfully employed in order to illuminate physical relationships. Thereby we obtain a more or less close interrelationship between the mathematicians, the physicists and the information processing specialists, corresponding to Fig. D.1. Mathematical systems serve for the construction of physical models, the numerical calculation of which is carried out today with electronic data processing equipment.

The function of the data processing specialists is primarily that of finding the most useful numerical solutions for the models which the mathematicians and physicists have developed. The feedback effect of data processing on the models and the physical theories itself is expressed indirectly in the preferential use of those methods for which numerical solutions are particularly easy to obtain.

The close interplay between the mathematicians and the physicists has had a particularly favorable effect on the development of models in theoretical physics. The modern quantum theory system is very largely pure and applied mathematics. The question therefore appears justified whether data processing can have no more than an effectuating part in the interplay or whether it can also be the source of fruitful ideas which themselves influence the physical theories. The question is all the more justified since a new branch of science, automaton theory, has developed in close cooperation with data processing.

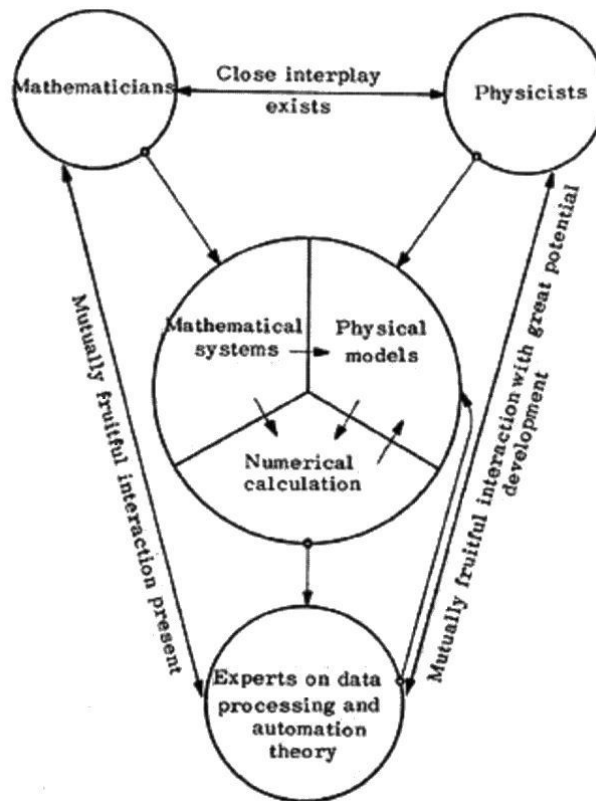


FIGURE D.1 –

In the following pages, several ideas along these lines will be developed. No claim is made to completeness in the treatment of the subject.

Such a process of influence can issue from two directions :

1. The development and supplying of algorithmic methods, which can serve the physicist as new tools by which he can translate his theoretical knowledge into practical results. Among these are included first all numerical methods, which are still the primary tool in the use of electronic calculating machines. The ideas expressed in the chapters which follow could contribute particularly to the problem of numerical stability.

Among these are symbolic calculations, which command an ever growing importance today. The numerical calculation of a formula is not meant by this, but the algebraic treatment of the formulas themselves as they are expressed in symbols. Precisely in quantum mechanics, extensive formula development is necessary before the actual numerical calculation can be carried out. This very interesting field will not be covered in the material which follows.

2. A direct process of influencing, particularly by the thought patterns of automaton theory, the physical theories themselves could be postulated. This subject is without a doubt the more difficult, but also the more interesting.

Therein lies the understandable difficulty that different fields of knowledge must be brought into association with one another. Already the field of physics is splitting up into specialized areas. The mathematical methods of modern physics alone are no longer familiar to every mathematician and an understanding of them requires years of specialized study.

But even the theories and fields of knowledge related to data processing are already dividing into different special branches. Formal logic, information theory, automaton theory and the theory of formula language may be cited as examples. The idea of collecting these fields (to the extent in which they are relevant) under the term “cybernetics” has not yet become widely accepted. The conception of cybernetics as a bridge between the sciences is very fruitful, entirely independent of the different definitions of the term itself.

The author has developed several basic ideas toward this end, which he considers of value to be presented for discussion. Some of these ideas in their present, still immature form may not be reconcilable with the proven concepts of theoretical physics. The goal has been reached if only discussion

occurs and provokes stimulation which one day leads to solutions, which are also acceptable to the physicists.

The method applied below is at present still heuristic in nature. The author considers the conditions not yet ripe for the formulation of a precise theoretical system. First of all, the existing mathematical and physical models will be considered in Chapter 2 from the viewpoint of the theory of automata. Several examples of digital models are presented in Chapter 3, and the expression “digital particle” is introduced. In Chapter 4, several general thoughts and considerations based on the results of Chapters 2 and 3 will be developed, and in Chapter 5, the prospects for the possibility of further developments are considered briefly.

ii Introductory observations

ii.1 Concerning the Theory of Automata

The theory of automata today is already a widely developed, and to an extent very abstract, theory about which considerable literature has been written. Nevertheless, the author would like to distinguish between the actual automaton theory itself and the thought patterns connected with this theory, of which considerable use will be made in the following chapters. A thorough understanding of the automaton theory is not necessary to an understanding of the chapters which follow.

The automaton theory appeared at about the same time as the development of modern data processing equipment. The design and the working method of these arrangements necessitated theoretical investigations based on different mathematical methods; for example, that of mathematical logic. The first useful result of this development was connection mathematics, in which particularly the statement calculus of mathematical logic can play an important part. Of particular importance is the realization that all information can be broken up into yes-no values (bits). The “truth values” of statement calculus assume only two ratings (true and false). The connecting operations and the rules of statement calculus can therefore be viewed as the elementary operations of information processing. Fig. D.2 shows the elementary connections corresponding to the three basic operations of statement calculus, conjunction, disjunction and negation.

Further research led to introduction of the term “state” of an automaton. In addition, input data and output data play a role. From input and initial

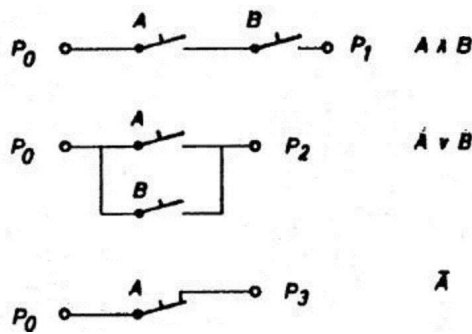


FIGURE D.2 –

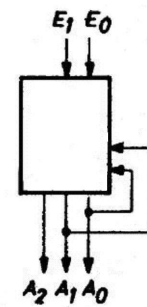


FIGURE D.3 –

state the new state and the output are obtained, corresponding to the algorithm built into the automaton. Fig. D.3 shows the schematic diagram of an automaton for a two-place binary register. In the figure, E_1 and E_0 represent the inputs on which a two place binary number can be entered and A_2 , A_1 and A_0 represent the outputs, which have the meaning of a three-place binary number. The two-place binary number formed from the figures A_1 and A_0 is relayed back to the automaton and represents the eventual state of the binary number. (In this case the states symbolize a number already entered into the addition process, to which the number E_1 , E_0 is to be added).

The algorithm given by the automaton can be represented by state tables in simple cases. These have the form of a matrix, and for every state and every input combination they give the resultant state or output combination. Fig. D.4 shows the state table for the automaton in Fig. D.3. In this particular case the state table represents an addition table. The theory of the automaton investigates the different possible diffractions of such an automaton and sets forth a series of general rules concerning its method of operation. It is important for what follows that the terms finite, autonomous and cellular automaton be understood. A finite automaton works with a discrete number of discrete states; it is roughly equivalent to a digital data processing machine, which is made up of a limited number of elements, each element capable of taking a limited number of states (at least two), with the result that the whole automaton can accept only a limited number of states. Similar conditions hold for the inputs and outputs. The autonomous automaton can accept no inputs (the outputs are

$E \backslash A$	00	01	10	11
00	000	001	010	011
01	001	010	011	100
10	010	011	100	101
11	011	100	101	110

FIGURE D.4 –

also relatively inconsequential). It can be represented, therefore, by a machine that operates independently, once started. Its states follow linearly in sequence, once the initial combination has been started, and the operational process cannot be influenced externally by the absence of one of the inputs.

The cellular automaton represents a special form of automaton built out of interrelated, periodically-recurring cells. This type of automaton is of particular importance for the observations which follow. Later it will be discussed in greater detail.

By the term “automaton theoretical way of thinking” we understand a manner of observation according to which a technical, mathematical or physical model is viewed from the standpoint of a lapse of states, which follow one another according to predetermined rules.

ii.2 About Computers

The automaton theory can be used as an abstract mathematical system, yet these thought structures can also be related to technical models, and similarly the automaton theory can be used for describing automata, particularly those suited for information processing. In current expanded usage, the term “compute” is identical with “information processing.” By analogy, the terms “computer” and “information-processing machine” may be taken as identical.

We distinguish between two classes of computers : analog computers and digital computers. In an analog computer, the steps in the calculation are performed in an “analog” model. Magnitudes representing numerical values are theoretically represented through continual physical magnitudes, such as positions of mechanical parts (torsion angle), tension, velocities, and the like. The machine operates essentially without end. The represented values lie obviously below certain technical limits. These are established by maximum values and by the accuracy of the system. The maximum values are given by a clearly-defined upper limit which corresponds to the technical limits of the system. In contrast, the accuracy has no clearly-defined magnitude, because it depends on change and on external influences (temperature, moisture, the presence of disturbing fields, etc.) One well-known analog computer is the slide rule. Fig. D.5 shows a mechanical adding mechanism the form of a lever which can be replaced with a rotating mechanism with gears, as in Fig. D.6. This mechanism is known in engineering by the inappropriate term “differential mechanism” and is employed in the rear axle of every automobile.

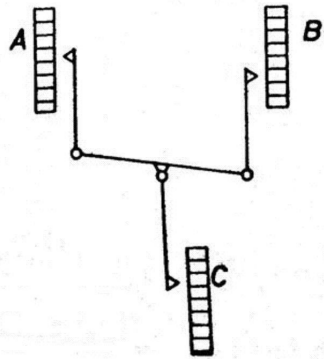


FIGURE D.5 –

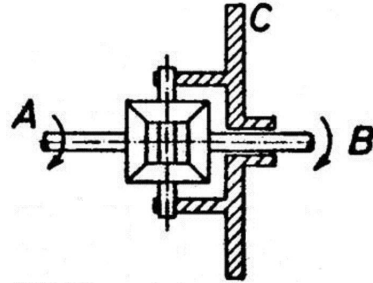


FIGURE D.6 –

A typical construction element of analog machines is represented by the integration mechanism shown in Fig. D.7. This operates with a friction disc *A* in contact with a friction wheel *B*. The distance *r* of the friction wheel *B* from the axle of *A* can be varied. In this way, the mechanism can be used for integration. In modern analog instruments, these mechanical elements are replaced by electronic ones. An integration can, for example, be carried out by charging a condenser.

Noncontinuous processes are generally not reproducible with analog instruments; in other words, analog computers are poorly designed for these processes.

With digital computers, all values are represented by numbers. Because a digital computer can hold only a certain limited sum of numbers, there is available for the representation of continuous values only a limited supply of values. This implies considerable divergence from mathematical models. Mathematical values are subject to the concept of infinity in two respects.

$$R d\psi = r d\varphi$$

$$\psi = \frac{1}{R} \int r d\varphi$$

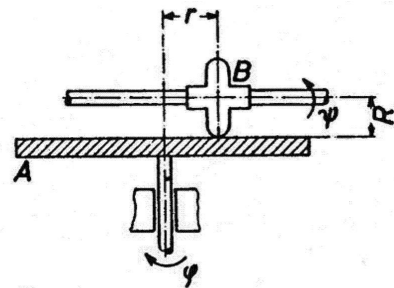


FIGURE D.7 –

First, the absolute magnitude of the numbers is unlimited; furthermore, between any two given values an infinite number of intermediate values may be assumed to exist. For this reason, computers have (independent of the number code employed) maximum values which, out of technical considerations (number of places of the register and storage), cannot be exceeded. In

addition, the values proceed in step-fashion. There are neighboring values between which no additional intermediate values may be inserted. This results in limited accuracy among other consequences. In contrast to the analog computers, the accuracy of digital computers is strictly defined and is not subject to any coincidental influences.

A further conclusion is that no digital computer can precisely reproduce the results of processes defined by arithmetic axiom. Thus, for example, the mathematical formula

$$\frac{a \cdot b}{a} = b$$

has general validity, with the one exception that a cannot be equal to 0. There is no finite automaton capable of reproducing this fact precisely and generally. It is possible, nevertheless, by increasing the number of places before and after the decimal point, for a digital computer to approach infinitely close to the laws of arithmetic.

We in the field of mathematics have already become so accustomed to the concept of infinity that we accept it without considering that every infinite term is related to a series expansion or to a limiting process ("for every number there is one which follows it"). By relating this process to automaton theory, we obtain in place of a static, predetermined, finite automaton a series of automata which are constructed according to a definite plan and differ from one another only in the number of places. The plan for construction of an automaton with n places is given; in addition, there are instructions for converting an n -place automaton to one with $n + 1$ places. By use of the limiting process $\lim_{n \rightarrow \infty}$ with the aid of series expansion the automaton rule for arithmetic operations is obtained.

The digital computer, because of its special ability to handle not only numbers but also general information (in contrast to the analog computer), has opened up completely new fields, discussed below in greater detail. In general, all calculation problems can be solved on a digital computer, whereas analog computers are better suited to special tasks. It must be stressed that digital computers work in a strictly determinative way. Using the same algorithm (i.e., the same program) and introducing the same input values, the same results must always be obtained. The limited accuracy always results in the same degree of inaccuracy in the results when an operation is performed several times on the same inputs. This is in contrast to the analog computer, in which the limited accuracy has a different effect each time the program is run and can be expressed only in terms of statistical probability.

By way of supplementary comments, it may be observed that hybrid systems have been developed which consist of a mixture of the principles of the digital and of the analog computer.

This can be simply carried out via a system in which the two computers operate side-by-side. They are joined by a digital-analog converter and an analog-digital converter (Fig. D.8). In systems of this type, the single parts of the problem are divided in such a way that the more appropriate device is chosen for each subdivision of the problem.

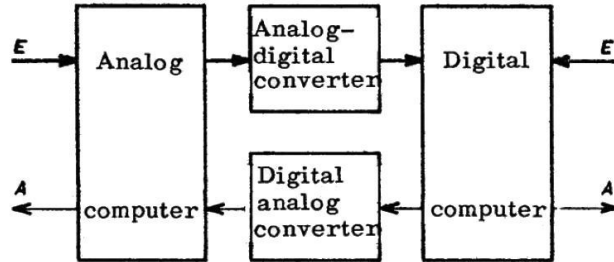


FIGURE D.8 –

The joining of the two systems can also be accomplished by the representation of the values themselves. Thus, for example, a magnitude may be characterized by the pulse density (Fig. D.9). Pulses themselves have a digital character, for they are normalized in intensity and duration; they are therefore digital, but their density (the number of pulses per unit time) can have any number of intermediate values, and it is therefore analog in character. A commonly-held opinion today is that the human nervous system operates on this principle.

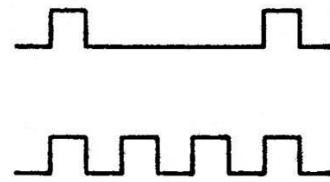


FIGURE D.9 –

ii.3 Differential Equations from the Point of View of the Automaton Theory

Observation of several differential equations reveals that this way of thinking is by no means self-evident to mathematicians and physicists. There are at our disposal a number of models of physical data, which can be represented by differential equations. For example, we can take a simple differential equation to represent the upper surface shape of a liquid in a rotating vessel, according to which at every point on the surface, the normal to the surface is determined by the vector sum of the gravitational and centrifugal accelerations (Fig. D.10).

This equation is written :

$$y' = \frac{r\omega^2}{g}$$

where ω is the angular velocity of the container.

The solution is very easy to obtain analytically :

$$y = \frac{\omega^2}{2g} \cdot r^2$$

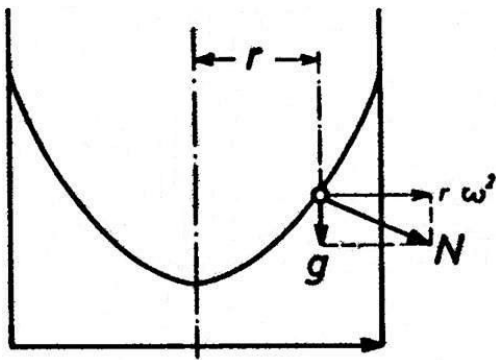


FIGURE D.10 –

In reality, we have here an expression valid for the situation only after equilibrium has been established. For every equilibrium situation there is an initiating action. In the experiment with a rotating vessel initially at rest, the rotatory motion must be transferred to the liquid through frictional forces. Only after complex wave interaction, which diminishes with time, will equilibrium be established. For this reason it is

not possible to describe the actual processes in this transition by means of our differential equation. The processes taking place during this period are considerably more complicated, and they are almost impossible to describe mathematically. We realize also that it is not necessary to follow each of these complicated processes when only the final state is of interest to us.

The relationships are very similar to many partial differential equations. These equations are used to describe the stress divisions of an equilibrium situation in plane and solid stress states. The establishment of equilibrium occurs in actuality *via* a highly complicated sequence of steps, in which once again the braking of these processes is the condition for the eventual establishment of equilibrium.

Differential equations describe only the final condition in the case of the theory of ideally incompressible fluids. The actual process leading to establishment of the end condition of equilibrium from a state of rest is hardly conceivable without taking compressibility and braking processes into account.

In the case of these differential equations, the issue is not one of a fundamental law, which can be described in terms of automaton theory as a

functional variable of different, sequentially-occurring states. This also has an influence on the possible numerical solutions. Differential equations which describe an allowed sequence of states of a system are often easier to solve numerically than those which represent no more than a control function over the final state. In fact, solutions for such end states must usually be found in a stepwise solution, often with help of a relaxation process. It is not necessary to attach value to the step-wise approximations of the final state in order to simulate natural or technical processes; thus, it is possible to apply mathematically-simpler processes in the approximation.

A differential equation which describes an evolutionary process from the point of view of the automaton theory may be called the “yield” form, because the following state arises from a given state through operation of the differential on the given state. In the case of liquids and gases, inclusion of the compression term leads first to this yield form. The state of a system is given by the pressure and velocity distribution. The differences in pressure result in forces leading to a new velocity distribution, which itself leads to a new density and therefore pressure distribution through the movement of the masses. The “state” of the field may be described, therefore, by a scalar density field γ and a velocity field v . The equation may be expressed in the yield form as follows :

$$k \text{ grad } \gamma \Rightarrow \frac{\partial v}{\partial t}$$

$$-\text{div } v \Rightarrow \frac{\partial \gamma}{\partial t}$$

(k is a factor which is determined by the physical conditions). The algorithmic character is even more clearly expressed in the following form :

$$v + k(\text{grad } \gamma)dt \Rightarrow v$$

$$\rho - (\text{div } v)dt \Rightarrow \gamma$$

Corresponding to the normal rules of programming language (algorithmic language), the same symbols on both sides of the yield sign refer to different sequential states of the system (v, γ).

In the case of incompressible fluids there is the condition $\text{div } \gamma = 0$.

This equation has no algorithmic character and cannot, as a result, be transformed into the yield form. It represents merely one condition for the correctness of a solution obtained by another means.

ii.4 Maxwell Equations

Maxwell equations can also be studied from this point of view. We will limit ourselves to those equations describing the expansion of a field in a vacuum :

$$\begin{aligned} \text{rot } H &= \frac{1}{c} \frac{\partial E}{\partial t} & \text{div } E &= 0 \\ \text{rot } E &= -\frac{1}{c} \frac{\partial H}{\partial t} & \text{div } H &= 0 \end{aligned}$$

Both equations, which contain the differential operator **rot** can be converted to the yield form easily :

$$E + c(\text{rot } H)dt \Rightarrow E$$

$$H - c(\text{rot } E)dt \Rightarrow H$$

(the rotor of H gives the increment of E ; the rotor of E gives the increment of H).

Both divergence equations, on the other hand, have no yield form. If the wave region of the field is taken into account we obtain :

$$\text{div } E = 4\pi\rho$$

This equation is not sufficient for the algorithmic description of the law of wave propagation. Are Maxwell equations therefore incomplete? They are used to describe the propagation of transverse, but not longitudinal, waves. The reason that Maxwell equations in their usual form are sufficient for the description of all processes occurring in electromagnetic fields rests on the fact that there exist in nature no growing, newly-appearing or disappearing waves. Only displacements of charge occur. With this sort of displacement, Maxwell equations are sufficient to describe the changes in fields associated with the displacements. The author has been unable to locate a precise mathematical proof of this in any text, but it must be assumed. An interesting

comment in this regard is found in “Beckersauter” (page 186), where the field for a uniformly-moving charge is developed. This results, interestingly enough, in elliptical deformation of the previously spherically-symmetric field. This deformation corresponds to the Lorentz contraction hypothesis. It is possible to reformulate the statement that “Maxwell equations are invariable in relation to the specialized theory of relativity” : “As a result of nature’s use of the trick of lateral expansion (rotor) in an expanding field, the system of the specialized theory of relativity is logically based”.

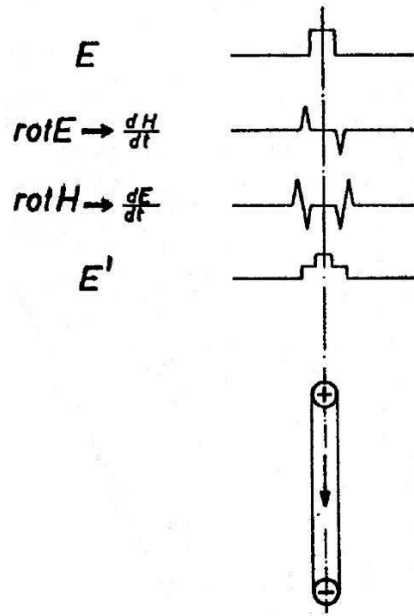


FIGURE D.11 –

We can conceive of the functional nature of this lateral expansion as follows : given that we want to calculate the field between two opposite charges $+e$ and $-e$, let us assume that we do not know the field distribution in itself well-known and also easily derivable. We begin, as shown in Fig. D.11, with a distribution sure to be false, by simply joining $+e$ and $-e$ by a linearly-constant force from the origin to the terminus. Application of the Maxwell equations to this field distribution results in a multistep asymptotic approximation of the field to be determined.

It is also demonstrated in this process that we obtain results without using the equation

$$-\text{div } E \Rightarrow \frac{\partial \gamma}{\partial t}$$

in the treatment of electromagnetic fields, although, as we have seen, this equation is necessary for the treatment of compressible fluids. We need not even introduce the electric field density γ . The fact that results are obtained without this term is not proof that nature works without resort to field density. Assuming that such a condition did exist, nevertheless, it would be nearly impossible to demonstrate its existence, for both “rotor” equations establish in themselves a field distribution such that

$$\text{div } E = 0$$

is generally satisfied. As a result, the divergent makes no contribution to the field distribution. Because it is impossible to create or destroy charges, we have no experimental means of testing the validity of the law of longitudinal expansion in nature.

What, then, is the rationale for examining this law? The question is interesting in connection with the concept of numerical stability, and it will be considered again below.

ii.5 An Idea about Gravitation

A short consideration of gravitation is introduced in this regard. If we accept the validity of the Maxwell equations, in their transmitted sense, for gravitation as well, then a simple explanation of the expansion of gravitational fields by moving masses and the invariance of the laws of celestial mechanics based on this distribution is applied to the special relativity theory. Because the relative velocity of the heavenly bodies within our observation range lie on the order of magnitude of $1/10,000$ of the speed of light, the gravitational magnetic fields were simply so weak that they were immeasurable. To be sure, small damping of planetary movements must be considered. The author would be very grateful for a critical observation of these thoughts by the physicists.

ii.6 Differential Equations and Difference Equations, Digitalization

If differential equations are expressed “yield” form, according to the automaton theory, then they can be simulated by a technical model (an automaton) and solved. In itself the analog computer is the ideal automaton. It works, at least in theory, with continuous values and operates constantly; in other words, we have a continuous flow of states, the latter of which is always determined by that which precedes it. In practice, analog computers are used primarily for calculation of differential equations. Nevertheless, there is a rather narrow limit to the capabilities of the analog computer. For partial differential equations, analogous technical models are available only under special circumstances.

The solution of differential equations with a digital automaton is immediately complicated by the previously-mentioned difficulties: differential equations operate with continuous values and infinite field densities. Digital

instruments operate with discontinuous values. An infinite field density would require an infinite storage capacity and infinite calculating time. Therefore it is necessary to reach compromises in both regards.

One normally proceeds from differential equations to difference equations when numerical solutions are sought. In this process, the values obtained are still regarded as continuous. In fact, the transition from differential equations to difference equations involves two boundary transitions : (1) $\Delta x \rightarrow dx$, and (2) enlargement of the number of places of the included magnitudes. The first boundary transition leads constantly to a limiting value which the second transition anticipates ; in other words, constructing difference quotients makes sense only if the gradations between values are much smaller than the chosen Δ -value. This fact has a definite influence on the numerical stability of a calculation.

If the transitions are carried out in such a way that the values remain of approximately the same order of magnitude as the step values, the staircase shape of the curve is maintained, and it is impossible to construct a differential quotient.

In the observations that follow, this distance will be utilized with design, specifically through consequential further development of the thoughts on digitalization.

Systematic narrowing of the number of places of the magnitudes being treated results in the limitation of variables to those encompassed by elementary logic ; for example, yes-no values or triply-variable values. As we will discover later, triple values and the trinary number system based on these values has certain advantages, since rounding up and rounding down are easier to carry out and the division by 6 necessitated by the division of the field area into 6 neighboring cells is also easier to calculate. By attaching the values $+1$, 0 and -1 to the numbers, this corresponds to the possible electrical particles $+e$, 0 , $-e$.

The continuous field density must be separated into single values for numerical solution, a process which is easiest with a grid. The simplest grid is doubtless an orthogonal one. There are other possible choices : the triangular and hexagonal grids in two dimensions, for example, and a grid in three dimensions corresponding to the most dense packing of spheres. If several different field values arise in the calculation (for example, velocity vectors and densities), it is not necessary that these values be localized on the same grid point. There is no need for the three components of a spatial vector to be localized. In this case, a division is possible as well. There is no further necessity in the construction of a digital space structure to approximate the

laws of Euclidean space. A number of general observations on the presentation of physical problems were presented earlier from the viewpoint of the automaton theory.

ii.7 Automaton Theory Observations of Physical Theories

Up to this point we have considered only the problem of using computers to approximate physical models and to follow physical processes numerically. It would be possible in this context to suggest a fundamentally different question : to what extent are the realizations gained from study of calculable solutions useful when applied directly to the physical models ? *Is nature digital, analog or hybrid ?* And is there essentially any justification for asking such a question ?

The classical models of physics are doubtless analog in nature. The field strength of different potentials, like the force of gravity, are not subject to a “particularization”. There are no such limits as “threshold values” (minimal size), limiting values (maximum values) or limits on the density of the field itself. Even the extension of classical laws by the theory of relativity is entirely within the conception of the continuum. Only for velocity is an absolute upper limit assumed to exist (that of the speed of light), and that concept is completely in accord with “analog” thought.

It was first with the introduction of the particular nature of matter through its subdivision into molecules, atoms and elementary particles that a few quantities assumed a discrete character, but this is not necessarily to be equated with “digital” interpretation of the laws of nature. The classical many-body problem was of an analog nature, even when each of the single bodies possessed individual characteristics with discrete properties (masses).

Quantum physics is the first to deviate in several respects from the concept of infinite quantities, to the extent that it assumes only discrete values for certain physical quantities. Best known is the relationship between frequency and energy of a light quantum, which is defined by the formula $E = h \cdot \gamma$, where h is a universal constant of nature. To be sure, the energy itself is not quantized, but only the quotient $\frac{E}{\gamma}$. This is somewhat different from the case where the energy can have only a discrete number of values because of the limited number of places in the calculator of a digital computer.

The postulates of the quantum theory have far-reaching consequences in relation to the quantization of different physical quantities. The conceptions

of the classical spatial continuum are being abandoned, it is true, but not through replacement of the continuum by a grid of discrete values, rather through a process whereby one moves to fundamentally different starting points, similar to a configuration room of higher dimensions, in which probability values are defined (for example, the probability of a particle being in a certain place at a certain time). Even in this concept the idea of the continuum is not rejected, for the differential equations of quantum mechanics are governed by no restrictions in relation to the magnitudes of fields.

The models of modern physics are concerned, therefore, both with continuous and discrete values. It would seem appropriate to consider a hybrid system. It will be extremely difficult to find a technical model of a hybrid computer which behaves according to the laws of quantum physics.

We have recognized the preliminary conclusion that our physical models may best be conceived of as hybrid systems. Can conclusions with respect to nature be drawn from this? Is nature itself therefore to be considered a hybrid system?

We have not yet disposed of completely digital physical models. If we are completely impartial, it appears a justified question whether infinitely-divisible quantities (in other words, really continuous quantities) have any reality in nature. What would be the consequences, for example, if we were to shift to complete quantification of all the laws of nature and were to assume in principle that every physical magnitude is subject to some sort of quantification?

Before an examination of the real question is attempted, let us examine first the classical model of thermodynamics, through which the relationship of gases is treated by the model of rubber balls moving freely through space and colliding with each other. If the static behavior of these balls is replaced by a differential equation, it is valid only for spatial dimensions that are large in comparison to the average distance between the individual particles. In effect, the model can be viewed as analog on a large scale, yet in detail it is characterized by the particle nature of matter.

What would the calculated solution look like, if we were to imitate directly the model of flying, colliding particles?

Of course, the starting point is no longer a differential equation; the flight paths of single particles are followed with digital calculations (Figs. [D.12](#), [D.13](#) and [D.14](#)).

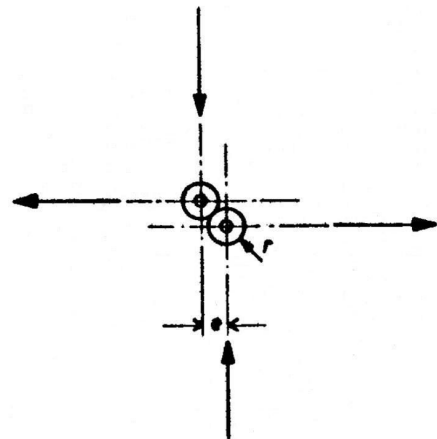


FIGURE D.12 –

It is quite simple for modern electronic computers to draw up a program for this purpose. We do not wish to become involved in these calculations in the course of our discussion (the calculation itself is relatively involved and boring) because a large number of particles are necessary for the results to have statistical value. The flight paths are simple to calculate, since they are rectilinear (gravity effects disregarded).

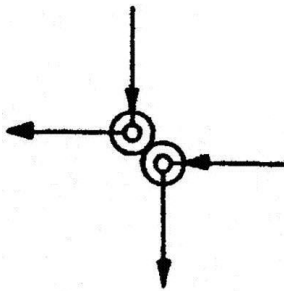


FIGURE D.13 –

The collision processes are the interesting part. Equal mass and equal elasticity of the particles is assumed. We shall first consider the case in which the particles meet exactly; i.e., first, that the paths lie in one plane and mutually intersect, and second, that the centers of both particles meet simultaneously at the point of intersection. This case is uninteresting, for the case of the elastic collision is not significantly different from that in which both particles continue undisturbed on their ways, if each particle is considered individually. Furthermore, in general situations, the probability of such a situation arising approaches 0 as the accuracy of the calculation is improved. Therefore, only those cases are of interest in which the paths do not exactly cross, or in which the centers arrive at the approximate intersection at only approximately the same time.

In this case the particles have different paths after the collision than before it. It is not necessary to stop here and establish the collision law firmly. The behavior depends on the size of the particles and the law of elasticity. Large particles collide more frequently than small ones. Hard particles behave differently than soft ones. The statistical result of the behavior of a large number of particles is the same. If we compare such a calculation model with the physical model, the following interesting aspects arise.

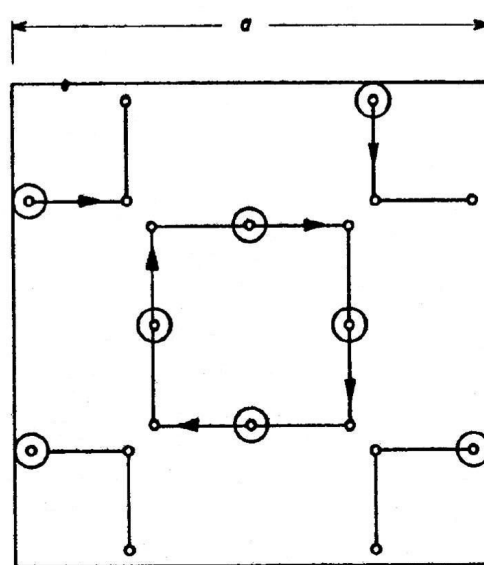


FIGURE D.14 –

In the case of both models, we can see that in general ordered states give rise to disordered states, or entropy increases. In any case, we can devise certain exceptional cases, for which a given entropy remains constant. Take, for example, a vessel with exactly parallel sides and a series of particles, the paths of which are exactly perpendicular to these walls and sufficiently far apart from each other that there is no mutual interaction of the particles. In this case, the paths remain unchanged in the sense of classical mechanics. This is also the case in the computer model if the coordinate system on which the calculation is based is set parallel or orthogonal to the walls. There are certainly other interesting special cases for which collision processes between the particles occur, yet nevertheless a certain ordering remains in force (Fig. D.14).

We are now aware that modern physics has replaced this classical picture. Collision processes between single particles are not precisely determinable, according to modern physics. There exist only the laws of probability, which correspond to the laws of classical mechanics, taken as a statistical average. Scattering is due to this effect, with the result that even for the theoretically-assumed special cases, the order of the system decreases with time and the entropy increases. How is this reproduced in the computer model? As long as we do not specifically program this scattering effect into our model, the carefully-constructed special case mentioned above does not exhibit any scattering effect. However, as soon as the system, through the introduction of a small scattering input, becomes out of step with the special ordering, the situation is similar to that obtained with the models of modern mechanics. It is not generally necessary to pay particular attention to scattering effects. The error inherent in the computation – special cases excepted – have the same effect (Fig. D.14). The classical model demands absolute accuracy in calculations, requiring in the computer model an instrument with an infinite number of places. Since this is not possible in practice, calculation errors enter into the collision processes, which have the effect – similar to the model of modern mechanics – that divergences from the paths predicted by the theories of classical mechanics appear. It would be possible to express these deviations by a statistical law. A significant difference does exist, however. In the model of modern mechanics the errors are real; in the computational model everything is strictly predetermined, not in the sense of classical mechanics but in the sense of defined calculating inputs, which can only approach the classical model. Both result in an increase in entropy.

The initially equivalent result (i.e., the increase in entropy) arises in both cases from the slight deviations from classical mechanics. In modern physi-

cal models, these deviations are defined by probability laws; in the case of computer models through defined calculation errors.

This may appear unimportant at first glance. Yet if we extend this thought process somewhat further, very interesting consequences in relation to causality may be drawn, which will be developed in Chapter 4.

Matrix mechanics can also be considered in the automaton theory. In any case, we need an automaton in which the transition from one state to the next is determined by probability laws. The transition matrices of matrix mechanics correspond to the state tables of the automaton. This possibility of automaton-theoretical observations will not be considered at greater length. In the next chapter, a few examples of digital treatment of field and particle problems will be presented.

iii Examples of digital treatment of fields and particles

iii.1 The Expression “Digital Particle”

Let us first consider one-dimensional space. In this regard we can relate an example from hydromechanics and one from counter engineering. Let us consider the behavior of frictionless gases in a straight cylinder. After eliminating and collecting terms that are irrelevant for our purposes (density, etc.), we can obtain a somewhat simplified relationship of the real physical forces.

We have two quantities : p (pressure), which we fix in discrete points 1, 2 and 3, and v (velocity), which we express in intermediate points 1', 2' and 3'.

$$\begin{array}{cccccc} p & 1 & 2 & 3 & 4 & 5 \\ v & 1' & 2' & 3' & 4' & 5' \end{array}$$

Δ_p^s and Δ_v^s representing the difference in p – and v – values between neighboring points, Δ_p^t and Δ_v^t corresponding to the differences between p and v in consecutive time intervals.

The following differential equations then hold :

$$\begin{array}{l} k_0 \Delta_p^s \Rightarrow \Delta_v^t \\ k_1 \Delta_v^s \Rightarrow \Delta_p^t \end{array}$$

Expressed in words : the change in velocity is proportional to the change in pressure and the difference in pressure is proportional to the change in velocity. In the second equation, the terminus Δ_p^t is converted in order to indicate that it refers to a Δ_p after that of the first equation. The two factors k_0 and k_1 , which contain the physical characteristics Δ_x (length component) and Δ_t (time component), can be combined for our purposes into a single factor k . We then obtain :

$$\begin{aligned} -\Delta_p^s &\Rightarrow \Delta_v^t \\ -k\Delta_v^s &\Rightarrow \Delta_p^t \end{aligned}$$

The sign $\rightarrow$ is used to indicate that Δ_p in the second equation is not identical with that in the first equation.

It is clear that these equations can be converted from differential equations to difference equations when Δ_x and Δ_t are allowed to approach 0. Exactly the opposite condition is of interest to us. Although mathematicians and programmers generally attempt to set up difference equations in such a way that the differential equation at the basis of the difference equation is approximated as nearly as possible, we are able to resolve the question by using the most general digitalization possible.

We are now able to convert a physical pulse law to an engineering counter law. If we let the quantities p and v and the corresponding values Δ_p and Δ_v assume only integral values, we must choose a whole number value of k in order for the difference equation to give whole number results. If we first let $k = 1$, we obtain the equations :

$$\begin{aligned} -\Delta_p^s &\Rightarrow \Delta_v^t \\ -\Delta_v^s &\Rightarrow \Delta_p^t \end{aligned}$$

We further attempt to assign p and v the smallest possible values, i.e. -1 , 0 and $+1$, and to study the behavior of the system that satisfies these conditions. We obtain as a result the following arithmetic relation :

$$\begin{aligned} v - \Delta_p^s &\Rightarrow v \\ p - \Delta_v^s &\Rightarrow p \end{aligned}$$

Fig. D.15 shows a simple calculating scheme for this rule. We have the four values v , $-\Delta v$, p and $-\Delta p$ per unit time. The spatial sectors are opposed to one another. Zero values are not written for purposes of simplicity. Four stable elementary forms are represented [(1), (2), (3) and (4)] which we will consider as mutually-independent “digital particles”. There are two time

units, t_1 and t_2 , respectively, for the values v , $-\Delta v$, p and $-\Delta p$; v and p are assumed for t_1 . It follows from this that $-\Delta v$ and $-\Delta p$ correspond to time interval t_2 and, following through the above equation, the values v and p correspond to the next time interval t_2 .

The equations relate to the traveling of a simple pulse. The particles are stable only at this velocity. At the same time, this velocity is the highest one possible for the system. The system permits no other velocities. Fig. D.16 shows a graphic version of this pulse.

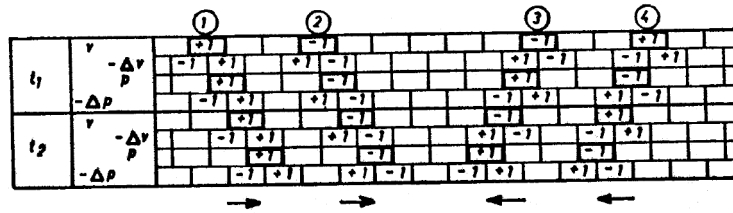


FIGURE D.15 –

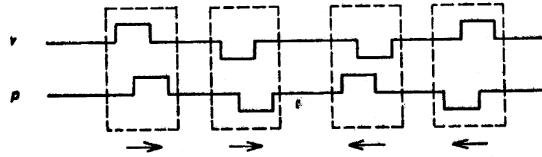


FIGURE D.16 –

From the standpoint of the automaton theory, we are concerned with a linearly-expanded infinite automaton which is repeated periodically in the automaton (cellular automaton). The v – and p – values represent the states of the automaton; Δv and Δp are derived from them. The above equation establishes the function according to which the subsequent state arises from the previous one.

Figures D.17 and D.18 show an instable form of expansion of an isolated pressure pulse, with which no velocity impulse is associated (as was the case in Figures D.15 and D.16). In Figure D.17, the Δ – values are omitted for reasons of generalization.

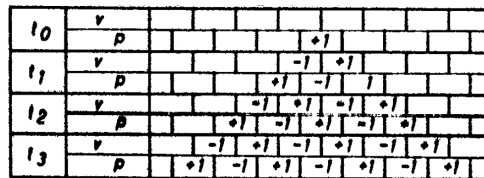


FIGURE D.17 –

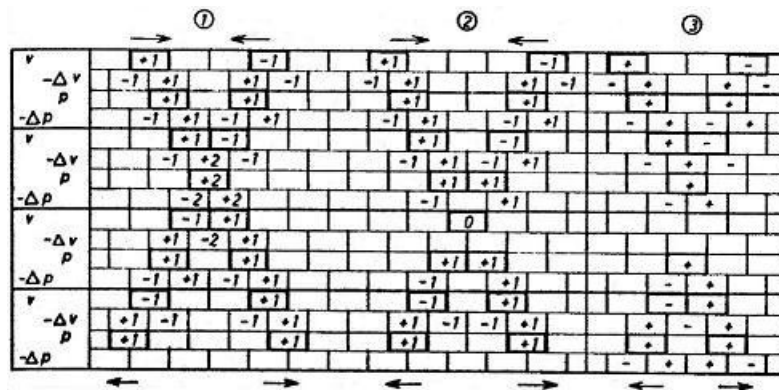


FIGURE D.20 –

ging shape. In a system strictly described by the superposition rule, there are no results possible which correspond to the reactions between elementary particles known in physics. This provides our evidence that it is not necessary to build linear elements into our models. The simplest and roughest form is general limiting of the values above and below. This may be demonstrated from the examples in Fig. D.20. Here we have two approaching digital particles, specifically in examples (1) and (2) on the left, corresponding to the previous reaction according to the superposition rule. We can see that in example (1), values +2 and -2 arise. In example (3), the particles pass through one another without values greater than +1 and -1 arising.



FIGURE D.21 –

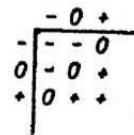


FIGURE D.22 –

In this situation, an interesting result of crude digitalization may be observed. The course of the collision process differs with the phase state of the distance between the two particles. This is not outwardly visible. Fig. D.20 shows example (1) with a limiting law corresponding to Fig. D.21. Here there are only three values : -, 0 and +. Fig. D.22 shows the relevant calculating system. It is constructed so that 1 + 1 results in a value of 1. We can see that in spite of this limitation, the particles are free to intersect one another, a result which in itself would not be expected at first glance, for crude curtailments of the calculating rule were made. Application of the calculating rule of Fig. D.22 to example (2) yields nothing new, of course, because in the

example the values -2 and $+2$ are not to be found.

It is interesting that in spite of this, a certain reaction process in particle interaction can be noted. If we consider examples (2) and (3), for example, it can be seen that in the case of (3), in contrast to (2), a certain retardation of the process may be observed. In (2) the particles intersect and proceed away from one another unhindered. In (3) we might argue that the particles first react with one another and that two new digital particles are emitted as a result of this reaction. The question as to whether (2) or (3) occurs is again dependent on the distance phase state and is outwardly a matter of chance. Without knowledge of the fine spatial structure, it can only be determined that in our example two fundamental situations are possible in particle interaction, for each of which the probability of occurrence is $1/2$.

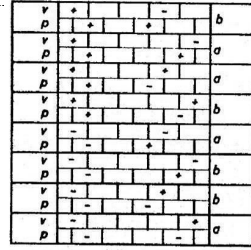


FIGURE D.23 –

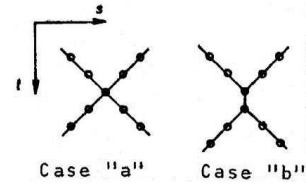


FIGURE D.24 –

Fig. D.23 shows a summary of the eight possible cases in particle interaction; Fig. D.24 represents the schematic, idealized particle paths for the two different interaction patterns a and b. It must be explicitly stressed that the paths are idealized particle paths. In reality, our model represents not continuous movement, but a process of stepwise progress.

It is interesting to note that in the nonlinear calculating rule (Fig. D.22), an isolated pressure point results in the emission of two particles (Fig. D.25).

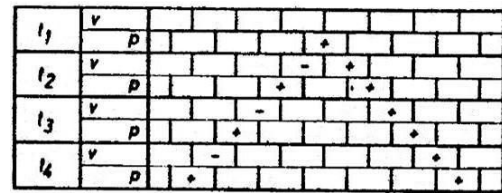


FIGURE D.25 –

Establishment of limiting values obviously sets limits on the free superposition processes. In the case of unlimited values, particles corresponding to Fig. D.15 are also theoretically superimpossible. That means that we can construct a pressure mountain of any height with its accompanying velocity distribution which satisfies the step-wise extension rule; i.e., which remains stable. These stable “larger” particles are always divisible into elementary particles. This is no longer true when the rule corresponding to Fig. D.22 is applied.

Our initial position, in which we have chosen the factor 1 relative to the Δ – value, corresponds to a very hard medium in the assigned physical pattern of a gas-filled cylinder. A more flexible situation is obtained when the factor is made smaller. In this case, nonintegral numbers arise in more accurate calculation. If we wish to continue with whole numbers or to introduce only minimal gradations, rounding up and rounding down must be introduced. In this respect also the ternary system is superior to the binary one. The value $1/2$ lies exactly midway between 0 and 1. The values $1/3$ and $2/3$ can also be precisely inserted between the values 0 and 1.

From there we want to make the following start :

$$\begin{aligned} v - \frac{\Delta p}{3} &\Rightarrow v \\ p - \frac{\Delta v}{3} &\Rightarrow p \end{aligned}$$

Values $\Delta_p/3$ and $\Delta_v/3$ rounded up or down to whole numbers. Fig. D.26 (1) shows a stable particle in this system with a period of $3\Delta t$. The velocity of propagation is $1/3$ of that of the particle in the corresponding figure (Fig. D.15). This corresponds as well to the physical model, in which a soft medium has a slower speed of sound. Here we have the situation that the “speed of switching” between neighboring particles is considerably higher (in the example three times as great)

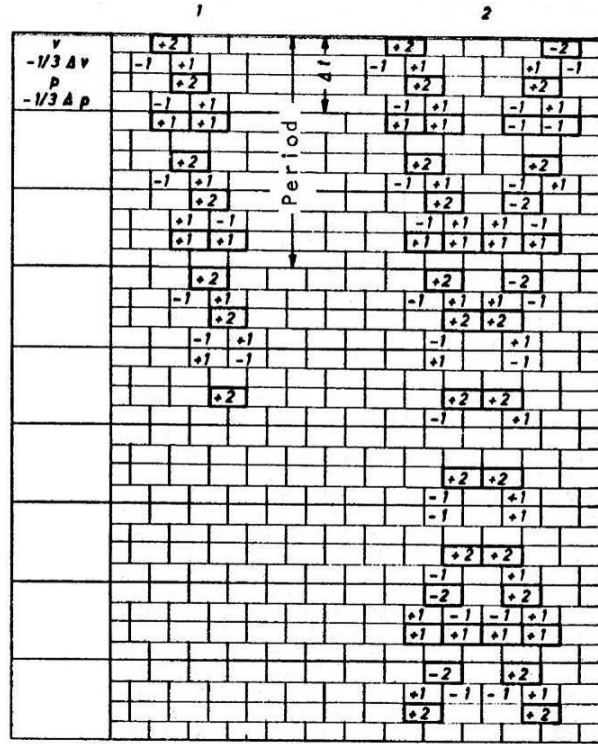


FIGURE D.26 –

than the particle velocity. In more complicated models of “calculating space”, it would be conceivable that speeds of light corresponding to maximum particle velocities, which are considerably slower than the speed of switching, exist. This does not mean, however, that in such a model “signal speeds” greater than the speed of light (in the model) are possible.

The speed of switching has a purely local meaning.

It is interesting that a digital particle assumes different configurations in the course of a period. The pressure pulse appears in part alone with a value of +2, in part as a pair with the values +1 and +1. The position of the particle is definable for the following period, but not without further information for the single phases of a given period. Is this not analogous to the quantum theory, which relates position and momentum through the uncertainty principle? In any case the computer model, in spite of the apparent error, is characterized by strict predetermined happenings.

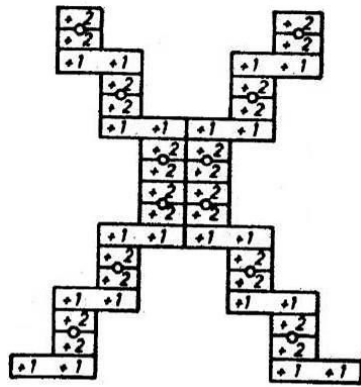


FIGURE D.27 –

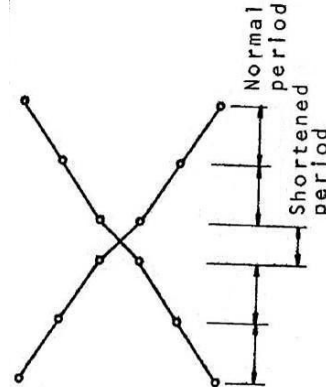


FIGURE D.28 –

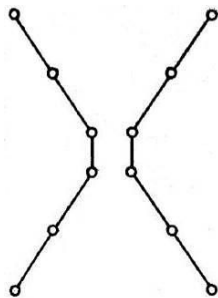


FIGURE D.29 –

Figures D.26, D.27, D.28 and D.29 show the process of interaction of two such particles, and more specifically Fig. D.26 (2) shows the detailed calculating scheme and Fig. D.27 an excerpt from it, in which only the p – values are represented, while Fig. D.28 shows the idealized particle path. The figures demonstrate that the particles do not simply pass beyond one another, but that they do react, this time with shortening of the interaction time (in contrast to Fig. D.24). The process can also be represented as one of repulsion (Fig. D.29). Here it may be seen in the mode of viewing the figures that terms like “passing through” and “repulsion” lose meaning when applied to the reaction of digital particles. The quantum theory has yielded corresponding results, although not in digital form.

In particle interaction corresponding to Fig. D.26, there are certainly many more differentiable cases apparent from systematic investigation, in comparison with the example from Fig. D.23. We must first investigate which

particles are possible in this system. The influence of the separation phases must also be taken into account, and finally the possibilities of the particles interacting in different phases must be considered.

It is not the purpose of this paper to carry out an exhaustive examination. The previous observation of a few simple examples stimulates a whole series of interesting concepts.

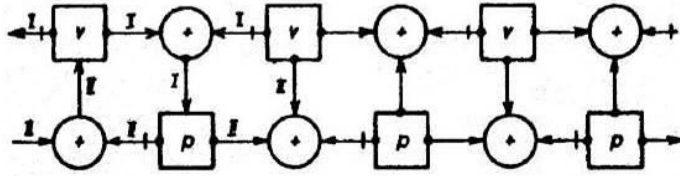


FIGURE D.30 –

Fig. D.30 shows the block diagram for a calculating space corresponding to the previously-introduced calculating rule. The

squares v and p represent registers to which numbers can be added. The shifting parts of the system, which serve to carry out subtraction, are represented by the circles marked with Δ . The vertical line at the exit of the Δ – members means negation. The block diagram can, of course, be subdivided into its single shifting elements. The symbols in current use reduce the shifting to its single elements, which correspond to the basic operations of Boolean algebra (conjunction, disjunction and negation). The three value information elements used here had to be converted to binary elements via two Boolean variables (2 bit). Out of 4 possible combinations of these two values, only three are employed. For this reason, a more detailed representation is omitted. In order to render the block diagram in Fig. D.30 operable, clean pulsing is necessary. Therefore the pulse beats are represented in Fig. D.30 by I and II. In this process it is taken for granted that the pure addition members work without time delay to build the Δ – values, while the registers transmit their information further only with the addition of the following pulse. This pulsing corresponds to the fine structure of the time dimension.

iii.2 Two-Dimensional Systems

Let us examine briefly the two-dimensional system. The simplest structure is a grid corresponding to an orthogonal coordinate system. The system possesses two definite axes which enter into even simple pulse propagation. We shall start with a simple rule, where every grid point can have the states 0 and 1. In every time interval one such 1 is transmitted to every neighboring grid point. The combination of pulses arising from different neighboring

points is carried out in accordance with the disjunction rule. If the state of the grid point (x, y) is $\phi_{x,y}$, we obtain the following equation :

$$\phi_{x-1,y} \vee \phi_{x+1,y} \vee \phi_{x,y-1} \vee \phi_{x,y+1} \Rightarrow \phi_{x,y}$$

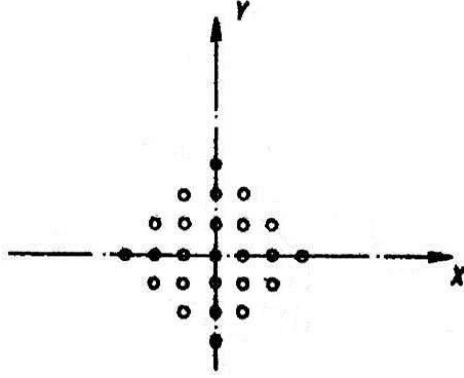


FIGURE D.31 –

transfer between the grid points the values are multiplied by a factor k . We obtain the formula for this rule :

Expansion along the coordinate axes is faster than along the diagonals. Little can be developed with such a rule, since after a short time it leads to a state in which all spatial points reach the state “1” and thereby no configurations, particles, etc. are possible (Fig. D.31).

Next we will consider a similar rule, in which nevertheless many-place values are allowed and combination occurs by addition. In the

$$K(\phi_{x-1,y} + \phi_{x+1,y} + \phi_{x,y-1} + \phi_{x,y+1}) \Rightarrow \phi_{x,y}$$

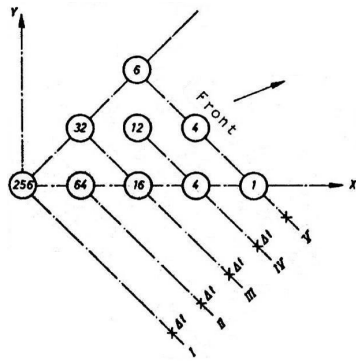


FIGURE D.32 –

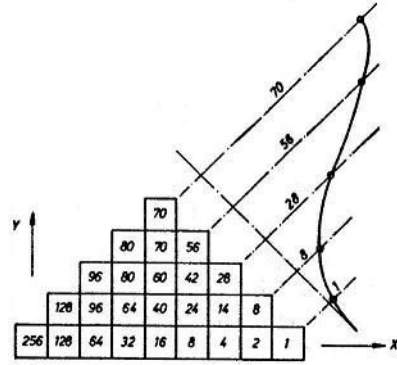


FIGURE D.33 –

Two examples for the factors $1/4$ and $1/2$ are given in Figures D.32 and D.33. For reasons of symmetry it is necessary to consider only a 45° section. As in Fig. D.32, the values are entered only for the front of the pulse. The roman numerals correspond to the individual time phases with a separation time of Δt . We can see from the examples that the front moves

as represented in Fig. D.31 ; i.e., with its peak along the coordinate axis, although the values along the diagonals are greater. The forward-rushing point very soon reaches its peak.

Because we cannot assume an infinite number of small values in digital space, the minimum value is soon reached ; i.e., the peak dies out. It would be interesting to follow the progress of such an expansion with the help of a calculating machine. The question of particular interest is whether and how quickly the values converge in a circular expansion pattern.

One thing is clear : it is impossible to construct digital particles from such a rule. We must find other rules.

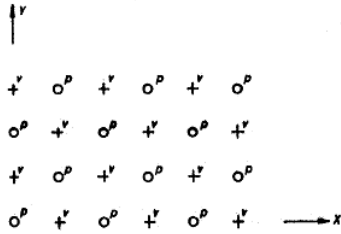


FIGURE D.34 –

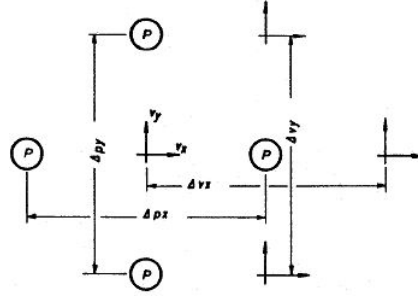


FIGURE D.35 –

It is possible to take the rules for linear space, which give rise to stable particles, and apply them to two-dimensional space. Of course, we then need an interrelationship of the two dimensions, for without it the single orthogonal grid points would have an independent existence.

Fig. D.34 shows one possibility of arranging the v – and p – values in a checkerboard. Fig. D.35 shows the individual values which emerge. Two components, v_x and v_y , must be considered for v . One value is sufficient for p . The two axes are coupled through p .

We can now formulate the following rule :

$$\begin{aligned} v_x - \Delta p_x &\Rightarrow v_x \\ v_y - \Delta p_y &\Rightarrow v_y \\ p - (\Delta v_x + \Delta v_y) &\Rightarrow p \end{aligned}$$

Because of the coupling through p , individual pulses corresponding to Figures D.15 and D.16 vanish. Stable, although not infinitely parallel, wave fronts can be built. Fig. D.36 shows such a wave front parallel to one of the coordinate axes, and Fig. D.37 shows a diagonally-moving wave. Fig. D.38

shows a propagation relation between the two waves. The propagation velocities are functions of direction.

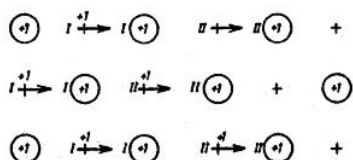


FIGURE D.36 –

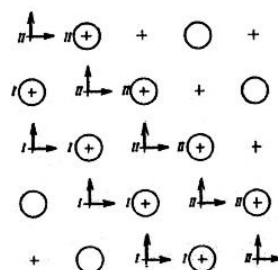


FIGURE D.37 –



FIGURE D.38 –

It would be interesting to consider the different consequences of more or less crude digitalization in this case. Because the rules are related to the equations of rarefied gas dynamics and hydrodynamics, it is interesting whether (for example) the hydrodynamically stable structure of a vortex can be crudely digitalized and “digital elements” can be constructed. This investigation can be carried out only with the help of calculating machines.

In order to construct stable particles in two-dimensional space, we shall first consider another manner.

iii.3 Digital Particles in Two-Dimensional Space

We shall assume an orthogonal grid pattern, corresponding to Fig. D.39. We no longer make the distinction between v - and p -points, but allow for each point the values p_x, p_y . For reasons of simplicity we first assume that the p -values can take on the values $-, 0, +$. We can then speak of p -arrows or of short arrows. First we establish that an isolated arrow (an arrow which has no perpendicular arrow arising at the same grid point) is directly transmitted to the next grid point. Fig. D.40 shows the four possible examples of this sort of single isolated pulse. It can be transmitted forward only in an orthogonal direction. We can first determine that there are two cases of interaction between two arrows approaching in the same orthogonal.

Both of these are shown in Fig. D.41. In one case, the arrows continue away from one another; in the other they cancel one another. Which case occurs depends on the separation phase. We still need a rule for the case of intersecting arrows. This is demonstrated in Fig. D.42. Two intersecting

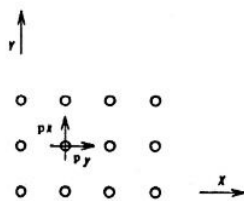


FIGURE D.39 –

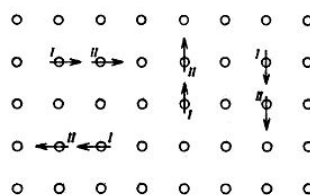


FIGURE D.40 –

arrows exist at point Z at time I . According to our previous rules, they would be propagated forward, each in its own direction, independent of the other. Now we establish that the two arrows are in fact propagated forward in their respective directions toward points B and C , and at points B and C they exchange direction. We obtain in this way a stable particle of period $2\Delta t$, which is propagated diagonally forward (Fig. D.43).

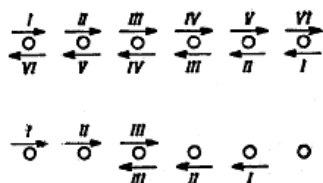


FIGURE D.41 –

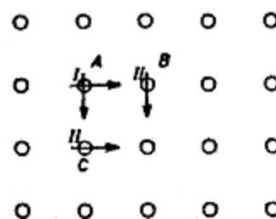


FIGURE D.42 –

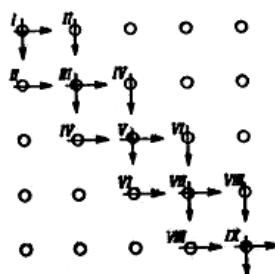


FIGURE D.43 –

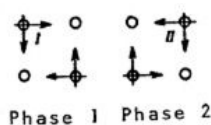


FIGURE D.44 –



FIGURE D.45 –

It is interesting to note that pockets arise from this rule which are fixed to 4 neighboring grid points; they have a period $2\Delta t$ (Fig. D.44). A doubly-stable pocket with period Δt is also possible (Fig. D.45). As may be seen from additional examples, these pockets cannot be destroyed.

We now have particles which can be propagated in eight discrete directions in a plane and standing pockets as well. Figures D.46-D.57 give a series

of interesting examples for the interaction of such particles. At first we shall maintain the condition that arrows may have only the values $-$, 0 , $+$. Two oppositely-directed arrows cancel one another at the same grid point, and two with the same orientation act as a single isolated arrow.

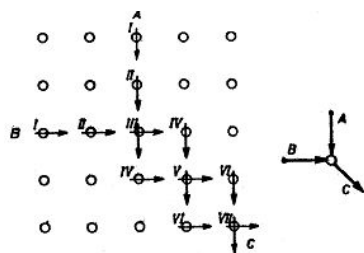


FIGURE D.46 –

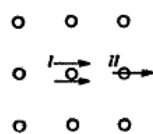


FIGURE D.47 –

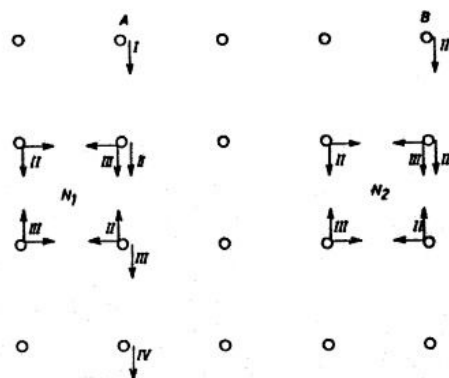


FIGURE D.48 –

It may be seen that the course of the different interactions is dependent on both time and separation phases. The particles can cross through one another, cancel one another or build new particles. Pockets are insidious because they can destroy particles without disappearing themselves. On the other hand, pockets can arise from specific forms of interaction (Figs. D.55 and D.57). In the model of a cosmos which functions according to this rule, all particles would eventually be converted into hard pockets. This model is therefore of little use.

In the interaction it is highly significant whether the point of intersection of the particle paths lies on a discrete defined point in the coordinate system. In this case a reaction occurs (for example, Figs. D.52 and D.53).

The possibilities of this system can be investigated by permitting the introduction of arrows of different absolute length. For arrows pointing in the same direction we use the addition rule. It is more difficult to expand the rule of Fig. D.42 to include two intersecting arrows of different lengths. We can reach the following agreement.

In the case of mutually-orthogonal arrows, the longer arrow is divided into two parts; the contribution of one is equivalent to that of the arrow orthogonal to it and combines with the first as in Fig. D.42. The remainder acts as an isolated arrow (Fig. D.58).

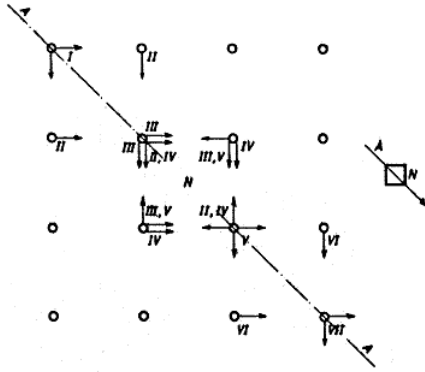


FIGURE D.49 –

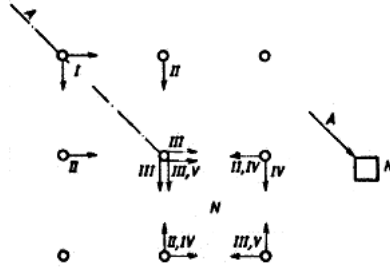


FIGURE D.50 –

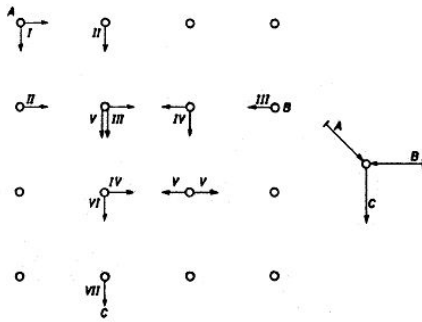


FIGURE D.51 –

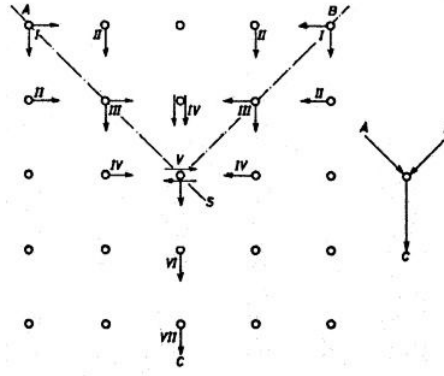


FIGURE D.52 –

We are now able to construct particles having different directions of propagation. The number of different directions possible is dependent on the number of values possible for the contribution of the arrow.

Fig. D.59 shows an example with a ratio of the arrows of 5 : 2. The direction of movement corresponds to the ratio of the arrows. The particles pass through different phases. The particle in Fig. D.59 has a period of $7\Delta t$. In the course of one period the particles pass through a discrete coordinate point Q (zero phase point). The particles “disappear” at intervals. It is possible to construct lines of the same phase (phase lines $\tau_0 - \tau_6$).

Fig. D.60 represents an example of the limitation of the possible discrete directions of motion. It must be stressed that there exists an interdependence between the velocity of propagation and direction. The chosen propagation rule permits no difference in velocity of the particles moving in the same direction.

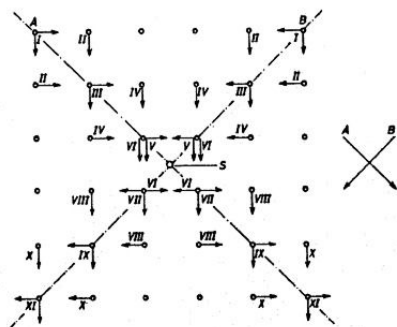


FIGURE D.53 –

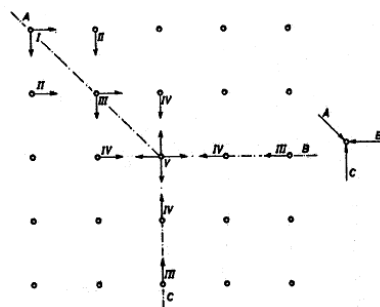


FIGURE D.54 –

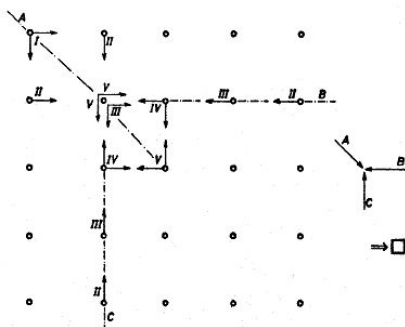


FIGURE D.55 –

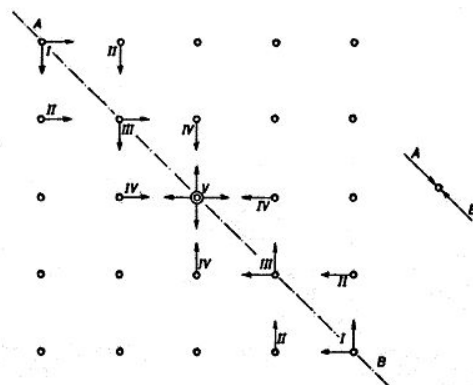
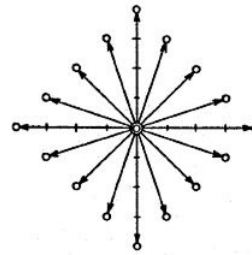
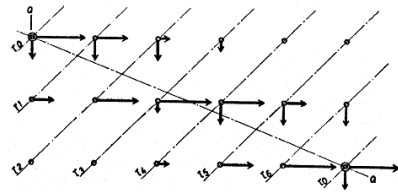
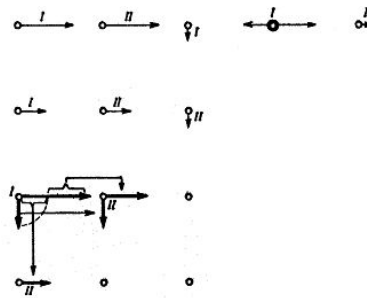
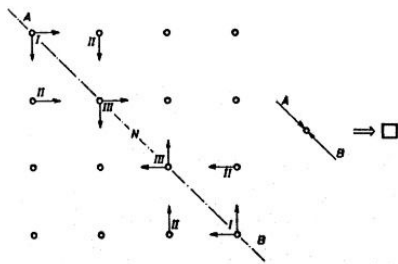


FIGURE D.56 –

Figures [D.61-D.66](#) show another series of interesting cases of interaction between such particles. Again the process of interaction is phase-dependent. A reaction between two particles always occurs, when they are respectively at the zero point at intersection (for example, Figs. [D.61](#) and [D.62](#)). But they can also react under other circumstances, as the examples in Figures [D.65](#) and [D.66](#) show. In these cases, the already – mentioned phase lines play a part. We could construct a time phase line R , which represents both particles. If this passes through the point of intersection of the particle paths S , a reaction is possible (Figs. [D.65](#) and [D.66](#)).

Of course, these examples are very simple and primitive. But even these simple forms yield an abundance of suggestions; they show that the basic method of digitalization adopted is of greatest interest and that development of the rules will yield additional concepts.



iii.4 Concerning Three-Dimensional Systems

The concepts developed in Sections 3.2 and 3.3 can also be applied to three-dimensional systems. The studies of the author are not yet complete in this area and should be reserved for further investigation.

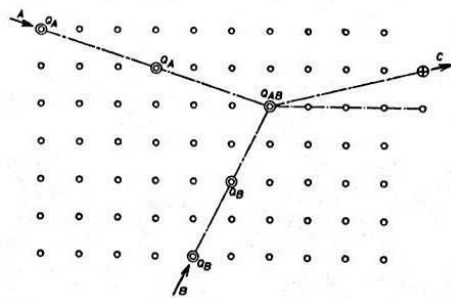


FIGURE D.61 –

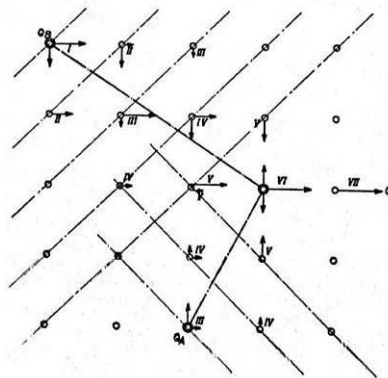


FIGURE D.62 –

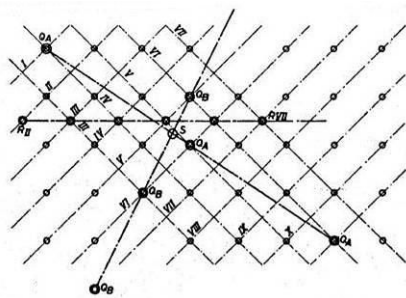


FIGURE D.63 –

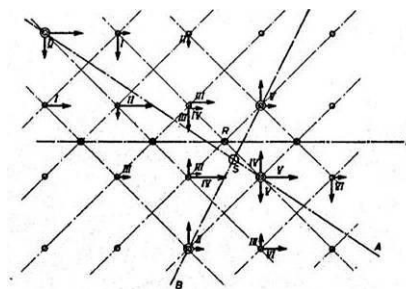


FIGURE D.64 –

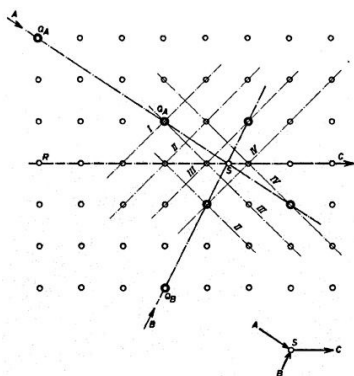


FIGURE D.65 –

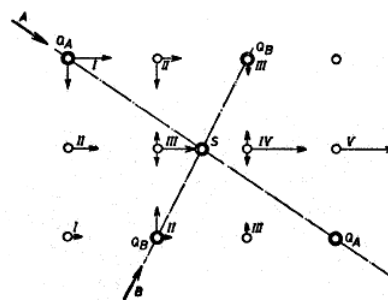


FIGURE D.66 –

iv General considerations

iv.1 Cellular Automatons

The examples of digitalization of fields and particles which have been presented are in their present unfinished form still far removed from being able to serve in the formulation of physical rules. Nevertheless, they give a rough impression of the possibilities for using the tools of the automaton theory to answer physical questions.

The examples have dealt primarily with point grids. A single cellular automaton consists, therefore, of a point grid which is bound to neighboring points through information exchange. In the cases shown in Figures D.34 and D.35, the grids are checkerboards of two different values, p and v , in grid form. There exist different possibilities for their combination, so that division into single automatons is not specific. This does not affect the behavior of the entire system.

In general, division of the continuum into discrete cellular automatons has different consequences, depending on the precise division. The idea of a grid spatial structure is already treated in various contexts by physicists, although not in regard to automaton theory. Generally speaking, the idea that the cosmos could really be subdivided into such cells is sharply repudiated by physicists. We agree that space cannot be viewed as a continuum even in infinitely small sections. The concept of a smallest length is already widely accepted today, while not in relation to the idea of subdivision into a point grid, but more as the principal limit in the differentiation of two different particles. The doubts relating to a grid structure are essentially as follows :

- (a) A grid structure would abolish the isotropy of space.

It is clear that a regular grid pattern establishes preferred directions. This has an effect, for example, in the expansion of fields (Figs. D.31, D.38) and in the discrete possible directions in which a digital particle can move (Fig. D.60). We know of no physical experiments which would provide a key to preferred directions of this type, but the field has not been systematically studied for this effect. Sober reflection reveals, nevertheless, that it is worthwhile to consider rules for a grid like spatial structure which do not allow the grid structure to become visible in regions of smaller and intermediate energy and frequencies. The grid constant must be assumed to be considerably smaller than the elementary shortest length of approximately 10^{-13} cm (Bopp assumes even 10^{-56} cm). The field of normal optics, for example, works with wavelengths of extraordinary length in comparison with these lengths.

It is hardly possible to think of an experiment which could determine the eventual discrete propagation direction of photons, when we assume the accuracy of such a change in direction (in circular measure) to be of the same order of magnitude that we are capable of differentiating between frequencies, namely 10^{-12} (Mössbauer effect).

Results of this sort can first be expected in the very high energy ranges, when wavelength and length of the period approach the grid constant. Only today do we have the capability to carry out such experiments. The author must leave it to the decision of the physicists whether and within which limits these phenomena could be observed with the aid of present experimental techniques.

(b) Curved volumes, as they are assumed from the general theory of relativity, are hard to represent with the grid structure of space. Bopp has chosen the expedient of assuming a Cartesian space in which the three spatial coordinates each converge on themselves. This can be imagined in two-dimensional space by assuming a toroid.

There are, of course, many possible deviations from these consequences. The whole subject is still too young for one to be able to draw final positive or negative conclusions. The following possibilities can be mentioned :

(α) The assumption of fixed circuits in the form of cellular automata is not the only logical possibility for defining logical connections between discrete values in space. If we introduce the change in the circuits as a function of the results of the previous process, variable circuits can be regularly developed.

(β) The concept of the growing automaton is closely related to the regular variability of circuits.

Both possibilities require at first a very well-prepared theory. Since automaton theory is a young field, the possibilities of which are in no respect exhausted, we can expect further developments in the direction being considered.

(γ) The assumption of a grid implicitly assumes that of an inertial system, which is contradictory to a strict interpretation of the theory of relativity. This will be considered at greater length.

In this light, the use of an orthogonal network is the most convenient way of beginning investigations. The results obtained in this manner will certainly be just as valid when in the course of time automaton theory yields new methods for use.

iv.2 Digital Particles and Cellular Automata

Digital particles may be considered as disturbances in the normal conditions of a cellular automaton. This disturbance has a distinct pattern which is subject to periodic changes. According to automaton theory, every state evolves from the preceding one ; nevertheless, the entire pattern can fluctuate in the process. To a certain extent we are concerned with “flowing states”. In accordance with this, digital particles can be regarded as “self reproducing systems”. A given pattern is generated in a neighboring region of the cellular automaton.

In the examples in Chapter 3, digital fields and digital particles are treated separately. Modern field theory takes pains to explain even elementary particles through singularities and special forms of fields. Automaton theory is understandably well-suited to digitalize such interpretations and to subject them to the rules of automaton theory. The author hopes to be able to treat this subject in greater depth in another contribution.

iv.3 On the Theory of Relativity

The question of the isotropy of space obviously requires coming to grips with the theory of relativity. The Lorentz transformations so important to the special theory of relativity, can obviously be infinitely approximated by numerical estimates. Nevertheless, it is very difficult to simulate in digital form the consistent form of the model of the theory of relativity. Our physical experience tells us immediately that no excellent coordinate system can be proven to exist, and that we are justified in considering each coordinate system to be as valid as the next one, in which case the Lorentz transformations formulate the relationships between these inertial systems. The strict interpretation of the special theory of relativity leads, however, to the conclusion that in reality no superior coordinate system exists, and that it is useless to search for such a system experimentally. In any representation of the cosmos as cellular automata, it is almost impossible to avoid the assumption of a superior system of movement. We can construct the structure of cellular automata in such a way that a greater number, although still a finite quantity, of superior coordinate systems are available. The constance of the speed of light in all inertial systems is represented by the digital simulation of the Lorentz transformations and the related shortening of bodies.

In any case, a relation between the speed of light and the speed of transmission between the individual cells of the cellular automaton must result

from such a model. These do not need to be identical. In contrast, it may be assumed that the speed of transmission from cell to cell must be greater than the speed of propagation of the signal obtained from this transmission. This greater speed of transmission has only a local meaning. Because of the anisotropy of the calculating space it is different in different directions, In any case the “digital” model, in comparison with the analog model of the relativity theory, yields a significant difference : the closer the velocity of the inertial system approaches the standard of the speed of light, the more critical the digital simulation of the processes becomes. In the case of energy-rich particles, we come to processes which can be characterized (at least to some extent) as a “miscalculation” of calculating space. In this way the essentially different behavior of particles of very high energy (high velocity, high frequency) can be explained.

A strict interpretation of the special theory of relativity has as a consequence that for every inertial system another one can be imagined, which moves with an initial velocity less than c . The physical rules are just as valid in the second system as in the first. This process can be repeated as often as desired, at least in principle. The complete monstrosity of this thought is only vaguely clear. Here it must be said again that every conception of infinity presupposes a limiting process. Here we are concerned with an infinitely frequent repetition of reaction of another inertial system which moves relative to the previous one. This process has a few consequences if observations of an information theoretical nature are applied, as we will consider in the following.

The following statement is also of interest.

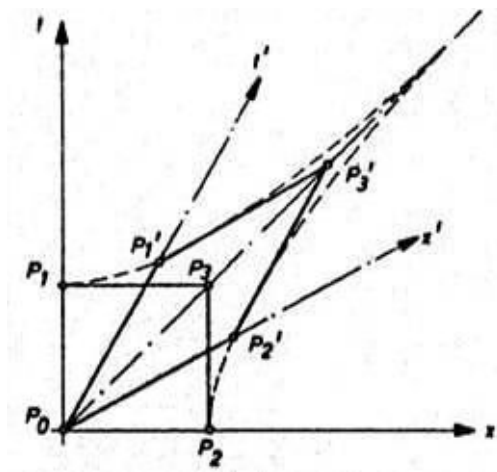


FIGURE D.67 –

We shall first introduce the term “shifting volume”. This is equal to the number of shifting parts involved multiplied by the number of shifting beats which take part in a given process, for example the period of a digital particle. Fig. D.67 shows a simplified representation, in which it may be assumed that a disturbance representing the digital particle extends for a distance $P_0 - P_1$. The particle is assumed to be stationary in the inertial system x, t . In this case, the space P_0, P_1, P_2, P_3 is equal to the shifting volume of a period. If

this particle moves relative to the system x, t we can speak of a second inertial system x', t , according to the special theory of relativity, relative to which the moving particle is stationary. The inversion corresponding to the Lorentz transformations yields the shifting volume P_0, P'_1, P'_2, P'_3 .

This is equal in area to the shifting volume P_0, P_1, P_2, P_3 . We can speak therefore, of *invariance of the shifting volume*.

iv.4 Considerations of Information Theory

The term information gains considerable meaning in the process of these different considerations. Information theory has formulated the term “information content” with clarity in regard to news-transmitting systems. For this reason, we are inclined to consider information theory as the theory of information processing. This is not correct, however. The easily accomplished application of terms from information theory in the neighboring field of news transmission unfortunately leads to frequent confusion. Even in the present observations we must be clear of our understanding of information content. It is difficult to speak of physical processes in terms of news transmission. This would be of interest in itself only insofar as we could include people in our consideration. If we assume an infinitely fine propagation of our news, transmitted through electromagnetic waves, it must be infinitely conserved, as long as limits are not established for them by the temporal finiteness of the universe. Metaphorically we can also consider the rays in the universe approaching us from other stars as news for people, in which case the question of the information content of this news makes sense.

Such a relationship between man and nature is to be found in the modern statement of the quantum theory, which attempts to relate all measurable quantities in a mathematical system. The information which we obtain from nature about the structure of atomic shells consists largely of the frequencies of the emitted light quanta. In this case, the use of the term “information content” is meaningful. The matter will not be further investigated here.

If we disregard this definition of information as the means of news transmission, it is still not possible to speak of information content of inhabited systems, if we consider the width of variation of the possible shapes of an object, a pattern or the like. Thus, a punch card may contain, due to its variability, a definite information content, measured in bits.

The technical characteristics of the punch card itself, including the accompanying punching and readout systems, set upper limits to the amount of information which can be entered, which is defined as information capacity.

In news transmission this capacity does not need to be completely used, so that the information transmitted from sender to receiver on the punch card can be below capacity.

It is also possible to speak of a maximum possible information capacity of a finite automaton, if we consider the number of its possible states as a measure. If this is equal to n , the information content is $\log_2(n)$ (logarithm to the base two). A programmed calculating machine represents this type of automaton, as we are aware. If such an instrument has m members, for each of which there are two possible positions (for example, flip-flops, ferrite nuclear rings in the storage, etc.), then the number of possible states is 2^m and the information capacity is equal to m . In this process no distinctions are made between the individual possible states. In the total of 2^m possible states every state is counted in which every register and storage unit is dissolved (i.e., set at zero) as are the states, as a result of which the solution of a very complicated differential equation is held in storage. Emotionally, we naturally tend to assume that the equipment contains no information in its zero state, although in the second state mentioned extremely interesting scientific results are available for use by mathematicians. This example shows the necessity for great caution in the definition of terms in information theory. The difference in this situation is that for the receiver, the two states have a fundamentally different meaning. The state of “everything dissolved” is only an extension of the receiver’s knowledge that the machine is in the ground state at the moment, while in the second case, the receiver’s knowledge is increased with regard to significant results.

If no account is taken of these individual values of information for the receiver, then the conclusion may be drawn that the information content of a finite automaton cannot be increased while running a calculation. Because the calculation is made completely automatically after introduction of the program and the input values, the results are established from the beginning. The results have greater value for the person using the equipment : for why would he let the computer perform a calculation if not to increase his knowledge, which is only possible if the final state of the automaton has a greater information content than the starting state.

The first result of viewing the cosmos as a cellular automaton is that the single cells represent a finite automaton. The question to what extent it is possible to consider the entire universe as a finite automaton depends on the assumption which we make in relation to its dimensions. If we take the toroid of higher order, as already suggested by Bopp, we are dealing with a finite automaton on the whole. It is originally valid that the individual cells can accept a limited number of states and have therefore only a limited

information content. This is equally true for the entire cosmos, if we make suitable assumptions about its limits.

Automaton theory demonstrates that different characteristic running patterns are possible for a finite automaton, several of which will be considered.

For every given state there is a succeeding state. It is therefore possible to express the relation “state A dissolves state B” as relation $F(A, B)$ and to represent it in the form of an arrow diagram. Such an arrow diagram is often called a “graph”. Figures D.68a-d show different types of arrow diagrams. It is important to remember that every state can have only one succeeding state, although there are several preceding states which can dissolve it. The process figures show that an autonomous automaton must end in a periodic cycle in every case, which under certain conditions can also degenerate into a single final state.

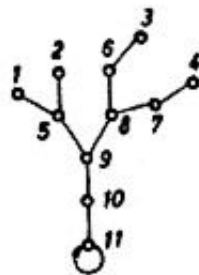


Fig. 68a.

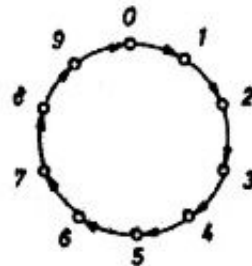


Fig. 68b.

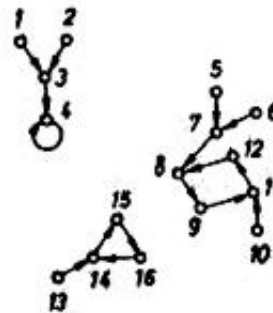
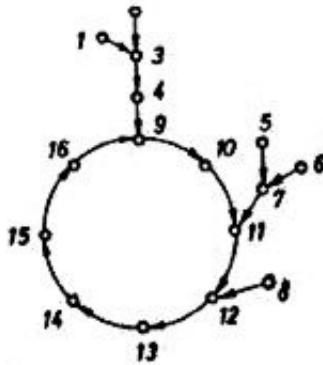


FIGURE D.68 –

This knowledge cannot be transferred to the individual cells of a cellular automaton, for they are related to neighboring cells through information exchange and therefore do not result in an autonomous finite automaton. In the assumptions of limits on the cosmos in the universe, we are concerned with a

finite autonomous automaton as soon as we exclude any sort of influences of a greater external world. The first result is the somewhat disillusioning consequence that the cosmic process must of necessity end in a periodic cycle. This realization, in itself logically unassailable, has other implications when examined quantitatively.

The dimensions of the universe are assumed to be on the order of magnitude of 10^{41} elementary lengths (10^{-13} cm) by some physicists (approximately 10 million light years). We are concerned therefore with a volume of approximately 10^{123} elementary cubes of the elementary length on a side. If an individual bit of information content is assigned to each of these elementary cubes, then we have already $2^{10^{123}}$ different states of the universe to consider. This number represents only a lower limit. In reality, a much finer grid must be assumed, for which it is not yet known how many variations at each grid point are possible. It must further be considered that space calculates extremely exactly. The relation of electrostatic interactions to those from gravitational fields is about $10^{40} : 1$. The interaction of nuclear forces are again orders of magnitude stronger. The higher of the two values represents in reality only a lower limit, which is most likely many orders of magnitude too small.

If we assume the number of time pulses to approach the order of magnitude of the spatial expansion, in effect 10^{41} , the result is obtained that in spite of this long time only a vanishingly small portion of the possible states of the cosmos can exist. There are $2^{10^{82}}$ types of reaction ways possible, each of which is independent of any other. This also means that the number of deflections and branchings is incomprehensibly great. The previously considered observations of automaton theory relating to Fig. D.68 lose all predictive value. Of what value is the realization that the evolution of the universe follows a periodic cycle, when even within the already very large range of time being considered one single period at most can pass, and most likely not even that?

The consideration of closed processes, i.e. of shifting processes involving a digital particle, appears more fruitful. We have already observed that a digital particle consists of a series of periodically-repeating patterns in a cellular automaton and that they are not fixed in position, but can move in the space of the single cells like the moving writing machine. The term “flowing state” was already introduced.

The question of the information content of a digital particle can be considered from several points of view. At first the digital particle accepts a set position in space at a particular point in time. The information content of

the digital particle cannot be greater than the information capacity of this position in space, which is determined by the sum of the possible states of this region. It is highly unlikely that every variation in state of such a limited region corresponds to a digital particle. It is much more likely that a limited selection dissolves individual stable period patterns.

We can inquire, entirely independent of the space associated with a digital particle, how many pattern variations representing phases of a digital particle are in fact possible? It is advantageous to classify the patterns along different lines :

- (1) type;
- (2) direction and velocity (pulse);
- (3) phase state;
- (4) position of the particle.

An answer to Question 1 assumes that we have at our disposal a model which permits different types of digital particles, as we have in nature with photons and electrons, etc.

An answer to Question 2 requires that our model accept different velocities and directions of propagation of the periodic pattern.

The phase sequence results from the periodic pattern sequence associated with the special type of particle and pulse.

Question 4 has meaning only when the interrelationship of the particles is considered. It is, of course, impossible for a closed region of space to hold the information about its own state.

The examples of Figures [D.42-D.66](#) from Chapter 3 satisfy these conditions only to a limited extent. First, the model permits representation of only one type of particle. Further only the direction may be varied, but not the velocity. The length of the periods of the individual particles is not constant, but this is not of interest to our consideration. The information content of this type of particle depends on the accuracy of representation of the arrow length or on the number of places with which it is digitally represented. If we assume absolute lengths of a component for Example 4, then we obtain 9 different arrow lengths, including the zero value, for that component; in two-dimensional space there are 81 different pulse variations. On the basis of these possible variations in the particles, even within the given limits it is possible to determine the information content of a particle. Each of these particles has a series of associated phase states, so that the number of pos-

sible patterns of digital particles is still greater. The particle in Fig. D.59 has, for example, 7 different phase states ($\tau_0 - \tau_6$).

The question of information retention in the reaction between digital particles is an interesting one. In the examples given in Chapter 3, pulse arrows are added in the course of reaction. This means that the number of places in the pulse arrow of the new resultant particle must be greater than the number of places in the reacting particle. If we eliminate arrow length of 0 for purposes of simplicity and assume that the arrow of the reacting particle can be represented by binary places, then the arrow of the resultant particle must be represented by 4 binary places. Before the reaction we have 2 particles, each of which has an information content of 2×3 bits (a total of 12 bits). After the reaction we have a particle with an information capacity of only $2 \times 4 = 8$ bits. During the reaction we have lost 4 bits of information. In this process we have permitted the arrow of the resultant particle to be represented by a greater number of places. This already means in itself the admission of a new type of particle. If this is not permitted, a rule must be found which takes effect whenever the permitted number of places are exceeded in the process of addition. If we simply assume that the maximum value may not be exceeded, then successive reactions lead after a certain period of time to the result that we are left with particles with the absolute maximum pulse arrows.

The examples chosen here for digital particles are still much too simple to be strictly related to physical processes. Actually, we are never confronted in nature with the situation that particles of the same type react with one another, not to mention the result that two such particles react to give a particle of a higher type. Conservation of energy, of pulse, of spin charge and so forth holds for elementary particles in physics. It is only when models of digital particles are at our disposal, with the help of which terms can be represented, that comparative observations with elementary particles in physics and their reactions are possible.

It is a question of obvious interest whether conservation of the different magnitudes cited in correspondingly-constructed digital particles is related to a corresponding conservation of information. The problem becomes even more complicated when fields are also considered. The author can only state the question without offering an answer to it. Perhaps the question is not so terribly important. Somehow the question amounts to the problem of "configuration", which is known to be very difficult to handle mathematically.

Here we come squarely into contact with one of the difficulties of information theory. In news transmission, the greatest possible information content

is obtained when the probability of the individual signals is distributed as uniformly as possible. This situation is referred to as the maximum entropy of information. It is easily possible to consider this in such a way that every possibility of relating previously-received news to the following symbol must of necessity diminish the information content, which limits through related redundancy the freedom on the selection of symbols (news, the content of which one can already predict, has no information content). Every sort of configuration necessarily represents through its rules a limitation of the possible means of representation and diminishes thereby the information content. Conservation of information and conservation of configuration are therefore contradictory to a certain extent.

The question whether or not tested terms in physics (energy, effective quantum, elementary charge, mass, etc.) can be interpreted by the terms of information theory or of information processing cannot yet be answered. In the model of a cellular automaton constructed so that processes occur in it which can be related to the listed physical quantities, these quantities must be represented by the construction of the circuits; i.e., by the values represented in the circuits.

Even more important than the term information content is that of information exchange. Something dynamic, not something static, results from circuit principles. Perhaps it could be called conservation of events or complication of events (Dr. Reche suggested the idea of “conservation of complicatedness”, although in another connection). Viewed this way, the shifting process acquires added meaning. If the effective quantum is assigned the dimension “shifting process”, we obtain the dimension “shifting process per unit time” for energy. The principle of conservation of energy can then be interpreted as the principle of conservation of events. The term “effective quantum” already points to a close relationship to shiftlike effects, namely the shifting process. The representation of energy as an “event” makes the relationship between energy and frequency more easily understandable. These thoughts are for the time being only simple speculation. Their purpose is to stimulate the application of automaton theoretical means of observation in physics.

A consideration of the Heisenberg uncertainty principle in the light of information theory follows. If a storage capacity of m bits is available for the digital representation of two quantities A and B , we are free to distribute the two quantities with different numbers of places and even differing precisions on the number of places. If n places are assigned to A , B has $m - n$ places. The error in A is on the order of magnitude of 2^{-n} , that of B the order of magnitude of $2^{-(m-n)}$. The product of both errors yields the constant 2^{-m} .

It is possible to assume that both conjugated quantities A and B are not directly represented by the pattern of digital particles, but represent derived quantities which appear only in certain processes. The limitations on the information content of the digital particles do not permit both quantities to be represented with the maximum possible accuracy. In the case of digital particles, even if one of the quantities is completely indeterminate, the other cannot be represented with ideal accuracy, but only with the maximum accuracy permitted by the limitations on the number of places. The following can be stated with regard to the principles of conservation : limiting values of the upper and lower sums must be considered. The laws of addition do not have unlimited validity. Similarly losses enter in the construction of models by falling below the threshold values. Digital models are possible in which, in spite of this occurrence, laws of conservation can be defined.

iv.5 About Determination and Causality

The question of determination and causality is closely related to observations from information and automaton theory. The expression “causality” is not strictly used in the literature. In the following it is always used to mean that which is generally referred to as “determination”, namely the definition of the succeeding state of a closed system as a function of the preceding state. The entire universe can be seen as a closed system, to the extent the necessary consequences of this assumption are taken into account.

Automaton theory works with the concept of the state of an automaton. Finite automata can receive a limited number of states. If there is no entrance signal, the resultant state results from that which preceded it because of the algorithmus basic to the automaton. Because automaton theory works with abstract concepts, this conversion from one state to the next occurs in theory without intermediate steps. Automaton theory does not ask the question exactly how this conversion occurs in an operating automaton. It is concerned solely with the fact that, for example, a flip-flop takes place from one state to another in the space of a certain time, the pulse time. The technological analysis of the turnover process, which is possible, lies outside the range of automaton theory observations, as long as it is not concerned with the comprehension of such details.

The opinion is held by some physicists, for instance Arthur March³, that direct conversion of an atom from one stable state to another is difficult to

3. For example, see March, Arthur : Die physikalische Erkenntnis und ihre Grenzen (Physical Perception and Its Limits), p. 19.

reconcile with the rule of causality. He understands the idea of causality in such a way that conversion from one closed system to the next requires a continual process. This interpretation can hardly resist the automaton theoretical consideration of physical processes. It cannot be assumed that this idea is based on reality. The process of thinking in whole numbers and in discrete states requires a thought process of non-continuous transitions, in which the law of causality is formulated in algorithms. Work with discrete states and quantification as such does not necessarily require rejection of the causal manner of observation.

This continuous transition in the sense of automaton theory must be differentiated from the thought of the continuous transition between the individual stable states of an atom. Since we are not able to analyze the process of such a transition experimentally, all theories on this subject belong to the realm of speculation. In the automaton theoretical sense, the natural objective is to create models which enable these transitions to be followed individually and permit explanation of the emission or absorption of photons in the associated process. We cannot predict whether this goal will ever be reached. The often-argued opinion that such transitions are essentially unanalyzable and that such experiments should be subordinated to more fruitful endeavors can, however, be refuted. Quantum physics provides statistical laws for such processes through which individual determinations are supplanted by statistical determinations. This subject will be pursued further in connection with the discussion of probability.

It is important to inquire whether the determination is valid in both time directions; i.e., whether later states of the system are clearly functions of the previous states as well as the reverse. The classical model of mechanics satisfies this demand for time symmetry ideally. Statistical quantum mechanics introduces the idea of probability and observes a deviation from time symmetry in the increase in entropy. In general, finite automata follow laws determined in only the positive direction. The algorithm establishes only which state arises from the given one, not the reverse. It is possible to construct automata in which the previous state is determined by the one which follows it, but this does not necessarily imply symmetry in the time direction. A consideration of computers may clarify this. A computer is – assuming unobjectionable work – determined in the positive time direction. In general, calculating processes are not reversible, which may be seen from consideration of the basic operations on which all higher calculations are based and which are not reversible (for example, $a \vee b \Rightarrow c$). A calculator is one example of a calculating machine which is effectively determined in both directions, because it counts forward in one time direction and backward in

the other, to the extent that we consider only the state tables and do not analyze the processes individually.

The different characteristic types of operation of an autonomous automaton were already discussed in 4.4 in connection with Fig. D.68. Type 68b would correspond to an automaton determined in both directions, as is the calculator mentioned.

A difference remains nevertheless : in the positive time direction, the rule by which the following state is related to the preceding one is explicitly given by the algorithm. In the negative time direction, there exists a single correlation, to be sure, but this correlation is only implicitly given ; i.e., it cannot be directly calculated without further knowledge. This difference is not clearly visible in the diagrams corresponding to Fig. D.68 and in the state table corresponding to Fig. D.4. In any case, this type of representation is possible only for very simple automata and serves more for primary experiments than for practical determinations of the automaton operation process. The actual rule for the formation of the following state from the preceding one is given by the automaton circuits. We are able to say that an autonomous automaton is determined in the positive time direction and that in special cases of negative time direction a “pseudodetermination” exists. The relationships of digital particles are similar in the cases discussed in Chapter 4.4. As long as such a particle follows its path independent of outside influences, a single sequence of states occurs. As soon as we consider the sequence of two particles, the conditions are immediately different. In this case, the examples in Chapter 3, Figures D.42-D.66 refer to irreversible processes. The basic shifting rule regulates the processes in the interaction of the particles, There is no sort of inducement for a particle to divide into two particles at any time. This statement makes only one assertion about the models used in Chapter 3. The question whether it is possible to construct usable models of digital particles which do not have this characteristic is difficult to answer. This is the same problem as the one confronting the physicist in the decay of elementary particles or atomic nuclei. The present state of theoretical physics is such that we can only give probability laws for such processes. In a model which follows a predetermined operation process and excludes working elements, in accordance with the probability laws, there are only two means of solution :

- (a) the digital model is constructed in such a way that it contains a sort of clock which dissolves the process when a certain state has been reached ;
- (b) the influence of the environment, (for example that of fields through which the digital particle moves) is taken into account. In the process of

moving through its different phases, a particle can pass through critical states in which the influence of the environment (frequency, etc.) causes particle division.

The present state of physical theories does not permit the drawing of conclusions about physical laws from these possibilities of digital models. What has already been said for the transition from one atomic state to another is equally relevant here : no experiment permits an examination behind the scenes, and all theories are essentially speculative in character. Nevertheless, it has been possible to determine a certain dependence of radioactivity at high temperatures, which corresponds to the assumption of critical situations influenced by the environment.

One result is important, in any respect : the assumption of valid determination only in the positive time direction is not influenced in the least by the dissolution of physical laws into the laws of probability. Similarly, the increase in entropy is not necessarily related to this question. From the viewpoint of automaton theory, each of these questions takes on another meaning. Entropy can be explained in a digital model, the operation of which is strictly determined.

Let us consider the classical model of physics from this point of view. As already mentioned, the validity of the determination, particularly in both time directions, requires absolute accuracy of the individual processes. It may hardly be assumed that serious considerations of the extreme significance of this assumption in regard to information theory have been made. Such a model requires an infinitely fine structure of spatial and temporal relationships. An infinite information content is required for an unlimited spacetime element. It is practically impossible to simulate such a model with computers because of the necessity of infinite number of places required. The sources of error are correspondingly great in the extremely large number of collisions between gas molecules, and these errors quickly lead to deviations from theoretical processes. This means that the better the causality rule is approximated in the reverse time direction, the more calculations we must be prepared to carry out in our model. This leads to the result that simulations of universal systems with causality functioning in both time directions belong to the category of “unsolvable” problems.

Of course, it can be said that this is true only for calculating simulative models. But this result should encourage us to reconsider the matter. Are we justified in assuming a model of nature for which no calculable simulation is possible?

From this point of view, it appears that the frequently advanced argument

of determination in both time directions should be fundamentally reexamined.

The question of time symmetry of the physical laws is frequently discussed in connection with the reflective characteristics of space. The observations of automaton theory might be of significant value in furthering this discussion.

iv.6 On Probability

The problem of determination in modern physics is closely related to the laws of probability. An observation from automaton theory may be inserted here. It is of course possible to build mathematical systems, such as matrix mechanics and wave mechanics, in which probability values play a significant part. The automaton theoretician can introduce the idea of probability into his theories and can establish a successive state dependent on probability values. To this point, the process is a simple mathematical game on paper. It becomes critical when we attempt to construct finished forms of such mechanisms which operate according to the laws of probability. Such calculations have been carried out in our calculating automata with considerable success for some time (Monte-Carlo method). The element of chance is introduced into the calculation in the form of “chance values”. The generation of these chance values is the decisive problem. There are two ways to accomplish this.

- (a) The values are generated by simulation of the dice method and that type of number series, in which no sort of dependence between the numbers exists. Such a number series can be developed from the calculation of irrational numbers (π , for example). In reality, this process is strictly determined. Nevertheless, we speak of pseudo-chance values. This process is completely sufficient when the generation rule for such chance values is carefully chosen.
- (b) A mechanism is taken from nature which is either so complicated that it cannot be shown to be regular or for which it can be said that, according to the valid laws of physics, it provides “real” probability values. The dice mechanism belongs to the first sort, where causal rules play a role but for which, in the case of a sufficiently carefully built die, equal probability for every case can be shown. The same is true for all games of chance (roulette, etc.). In the other case we rely on the fact that, for example, the radioactivity of a certain material is subject to strict probability laws. Whether the probability process is in reality determined in these atoms is not significant, for experience shows that

in any case the laws of probability can be assumed without leading to incorrect results. In this case the calculating automaton regards the probability values to a certain extent as external input values. It remains true, however, that real probability values are hardly possible in technical automata.

It must also be remembered that the choice of algorithm for creation of the pseudo-chance values is highly significant in Case (a). This means that only those choices from the range of basic number series are possible which follow one another as irregularly as possible and which have the most uniform possible distribution of probability. This means that longer series of the same number and series of numbers in the same separation (1, 2, 3) must be excluded, although these series are just as probable or improbable in real series of chance values as any other number series.

Of course, we can ask the purely speculative question whether true probability laws are admissible to automaton theoretical observations of physical processes. This question is a philosophical one, and is only noted here, without an answer.

iv.7 Representation of Intensity

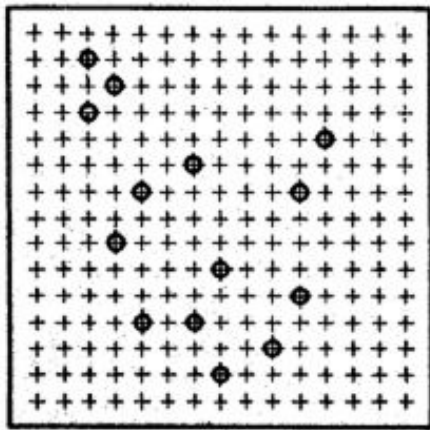


FIGURE D.69 –

The representation of intensity of field strengths and other numerical quantities in cellular automata must be specially considered. For this reason, a few basic possibilities are considered here.

Fig. D.69 shows a two-dimensional grid in which individual grid points are occupied by elementary logical values; for example, yes-no values. If we assign to these values the numbers 0 and 1, the statistical distribution of the 1 values represents a scale for field strength.

This sort of representation can accomplish little, of course, if many orders of magnitude of density must be taken into account. As already mentioned, the relationships of electrostatic interactions to gravitational interactions is on the order of $10^{40} : 1$. If we wanted to represent these intensity differences in a three-dimensional space

corresponding to Fig. D.69 using yes-no values, a cube with a side length of approximately 10^{13} grid units would be necessary. This represents only a lower limit, for in reality field strengths can differ by even greater orders of magnitude. If we take a grid with the elementary length of 10^{-13} cm accepted by physicists, it would mean that a space of many cubic centimeters would be necessary, according to these calculations, to represent the field intensity. This type of model cannot be very useful, entirely independent of the fact that it is extremely difficult to establish laws for stable digital particles with this sort of statistical distribution.

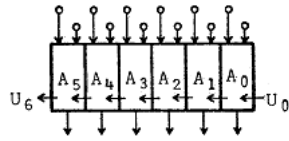


FIGURE D.70 –

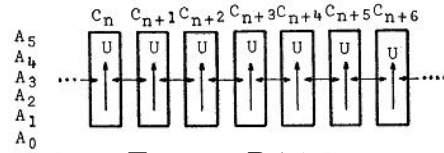


FIGURE D.71 –

A much more rational method is offered by the principle of place values. This does not lead to the idea to construct calculating automaton according to the principle of Fig. D.69. Fig. D.70 shows the ideal arrangement of an adding machine consisting of neighboring cells and among which a hierarchical ordering is seen. The individual cells are coordinated with numbers of different value. This is reflected in the one-sided construction of the transmission process $u_0 - u_6$.

Fig. D.72 shows the transmission of this thought process to a linear cellular automaton. Each cell is allied with a complete adding machine. Each cell C_i is subdivided into the individual addition steps $A_0 \dots A_5$. In the construction of such a shifting system it must be remembered that the transmissions among levels within the cell must be coordinated in time with the transmission of information between the individual cells.

This principle is relatively easy to put into practice for one-dimensional and two-dimensional cellular automata. Theoretically it can be applied to three – and more-dimensional automata without any modifications. In addition to the dimensions, which correspond to the topological arrangement of neighboring cells (space dimension), there is also a level dimension. This is only imaginable in three-dimensional space and must be constructively built into (projected into) three-dimensional space.

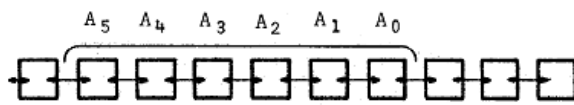


Fig. 72a

The further question can be asked whether in a symmetrically built cellular au-

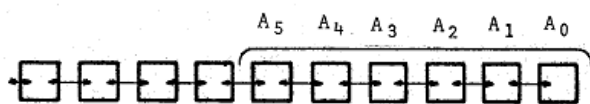


Fig. 72b

tomaton a hierarchical ordering can be introduced by the manner of occupancy. Fig. D.72 demonstrates the principle. The single cells can contain, for example, single addition steps and are not able to accept several-place numbers. These are divided among several neighboring cells, according to the place value principle. The difficulty arises in the fact that this sort of arrangement is of the nature of occupancy. If the concept is applied to a several-dimensional automaton, it is easy to see that major complications develop.

Cellular automata provide an elegant solution when each cell contains a complete calculating system, as symbolically represented in Fig. D.73. These single calculating systems contain both information-processing and information-storing elements.

The net automaton represented in Fig. D.74 is a further development of the cellular automaton corresponding to Fig. D.73. The individual cells are responsible here for only information processing. Branching lines B connect the individual cells and serve both for information transmission and for information storage. The individual cells can consist of single-place adding units, according to the series principle valid for calculating machines. Preliminary investigations by the author have shown that this type of automaton is highly successful, specifically in the solution of numerical problems as well as in simulation of physical processes. More specific consideration will be the subject of another paper.

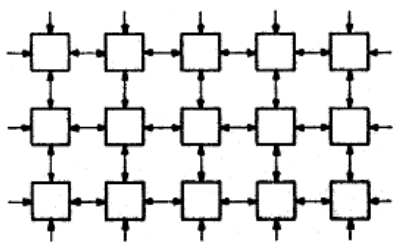


FIGURE D.73 –

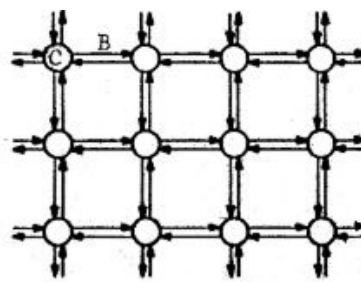


FIGURE D.74 –

v Conclusions

Even if these observations do not result in new, easily understood solutions, it may still be demonstrated that the methods suggested have opened several new perspectives which are worthy of being pursued. Incorporation of the concepts of information and the automaton theory in physical observations will become even more critical, as even more use is made of whole numbers, discrete states and the like.

A relating of different possible conceptualizations is attempted in the following table :

CLASSICAL PHYSICS	QUANTUM MECHANICS	CALCULATING SPACE
Point mechanics	Wave mechanics	Automaton theory Counter algebra
Particles	Wave-particle	Counter state, digital particle
Analog	Hybrid	Digital
Analysis	Differential equations	Difference equations and logical operations
All values continuous	A number of values quantized	All values have only discrete values
No limiting values	With the exception of the speed of light, no limiting values.	Minimum and maximum values for every possible magnitude
Infinitely accurate	Probability relation	Limits on calculation accuracy
Causality in both time directions	Only static causality, division into probabilities Classical mechanics is statistically approximated Based on formulas	Causality only in the positive time direction introduction of probability terms possible, but not necessary Are the limits of probability of quantum physics explainable with determinate space structures? Based on counters

In view of the possibilities listed, it is clear that there are several different points of view possible :

- (1) "The ideas of calculating space contradict some recognized concepts of present-day physics (for example, space isotropy); therefore, the fundamental basis must be false."
- (2) "The laws of calculating space must be revised with the object of eliminating the existing contradictions."
- (2) "The possibilities arising from the ideas of calculating space are in themselves so interesting that it is worthwhile to reconsider those concepts of traditional physics which are called into question and to examine their validity from new points of view."

The author has greatly enjoyed being able to discuss this subject with a few mathematicians and physicists. The greatest handicap to cooperation is certainly the difference in terms between the individual, specialized fields of knowledge. We hope that this chasm will be bridged in time and that through cybernetics, a true bridge between physics and the automaton theory can be built.

Independent of the possibility that the idea of calculating space can be directly applied to physical determinations, there remains the major task of providing theoretical physics with an aid in calculating and of finding numerical solutions to very complicated relationships. In spite of the use of huge computers in the field of physics, the applications of "software" in physics are still much more limited than the applications of "hardware". With huge accelerators that cost hundreds of million dollars we are able to obtain particles of very great energy, requiring a fundamental reexamination of the general validity of our basic theoretical hypotheses. Is there not a considerable danger that the software lags behind the hardware of physics, and that we will soon be unable to evaluate the determinative results of our practical experiments?

In the field of information processing we are already spending equivalent amounts on hardware and software. In physics the ratio of expenditures is probably between 1 :20 and 1 :100. The result in chemistry is about the same. Although the laws of electron shells have been generally known for a long time, young scientists are able to explore them only within circumscribed limits in precise, analytical chemistry. The author hopes that the ideas of calculating space after a period of adaptation will be of assistance. The first step would be further development of the models of the automaton theory

approximately along the lines suggested in this article. When this process has reached a certain maturity, then specific goals can be set.

It must be stressed that the experiments of the author are confined to pen and paper experiments. Further experimentation must be carried out with the help of modern computers.

Appendix E

Afterword to Konrad Zuse's Calculating Space

Of the version of *Calculating space* rewritten and reedited with A. German (School of Informatics and Computing, Indiana University Bloomington, U.S.)

There are many parallels between Zuse's and Turing's interests. In the mid 1930s, some researchers were engaged in what amounted to an inquiry into the nature of computation, and trying to figure out whether it would be possible to build a computing machine. In part this was a consequence of Hilbert's programme, but it was no doubt also due to a certain chain of historical events. As pointed out by Raúl Rojas¹, people started to think about computers when it was time to build computers. There were of course Schönfinkel (SKI combinators), Church (λ calculus), Post (tag systems), Kleene (recursive functions), Turing (a-machines), among a few others.

Perhaps the main difference between all the other approaches and Zuse's lies in the fact that Zuse was a civil engineer aiming to solve concrete problems and, as such, his approach was quintessentially practical. Thus Zuse's goal was from the beginning that of building a concrete, mechanical realization of computation. Turing's approach falls mid-way between the purely abstract and a practical realization. This fact alone may explain why Turing's work was, in the end, more visible than others. Zuse's approach being an engineer's answer to the question of computation, his solution took the form of an actual machine².

1. In a recent talk *Zuse and Turing in Context* in Cambridge, UK on February 18, 2012.

2. The most comprehensive source of information is the Konrad Zuse Internet Archive curated by Raúl Rojas available online at <http://www.zib.de/zuse/home.php> (accessed in April 2012). His son, Horst Zuse, maintains his father's homepage, available at <http://>

Zuse may not have realized that there was a fundamental concept behind the question all these people were asking and ultimately trying to answer (Zuse was working in relative isolation, unlike the others, who for the most part knew of each other). Turing finally provided the closest answer to the question with his concept of computation universality, the founding notion of Computer Science. Paradoxically, today's digital computers may be more similar in some respects to Zuse's than to Turing's idealization, certainly because Zuse had to deal with the minutiae of actually building a physical machine (for ex., the IEEE Standard for floating-point coding is almost the same as the representation used in Zuse's Z1 and Z3). Zuse never thought of universality as Turing did, but as Rojas has proved, not without some creativity, the Z1 and Z3 *accidentally* (because it was never Zuse's purpose, and he didn't even formulate the question) turn out to be capable of universal computation⁴. Zuse never thought about how the machine could get into an unbounded computation (necessary for universality), for example, and if it did, how to make it stop (Rojas suggests that there would have had to be a mechanical/electrical hack to arbitrarily stop the machines, with the required computation finished and somehow encoded among other computations in the output, if unbounded computation were allowed – by, for example, looping a punched card).

Upon graduating in 1935, Zuse became a stress analyst for the Henschel Aircraft Company, where he worked on problems of aircraft vibration. Stress analysis involved formidable calculations, which at the time could only be performed with great difficulty using teams of human “computers” equipped with desk calculating machines.⁵ Zuse thought that many of the calculations he was performing could simply be automatized. With a 1936 research grant from the *Reichsluftfahrtministerium* (the German ministry of aviation), he coincidentally built his first computing machine between 1936 and 1938, and in 1938 he was building his second one, using phone relays unlike the first one, which was mechanical. His Z3 was completed in 1941, was fully operational, and was able to perform calculations⁶. His Z1 was already programmable

www.horst-zuse.homepage.t-online.de/konrad-zuse.html (accessed in April 2012). And Juergen Schmidhuber³ also maintains a website devoted to Zuse, available at <http://www.idsia.ch/~juergen/zuse.html> (accessed in April 2012).

4. See Raul Rojas' “The Architecture of Konrad Zuse's Early Computing Machines,” in “The First Computers – History and Architecture,” MIT Press, 2000, pp. 237-262, edited by R. Rojas and Ulf Hashagen.

5. <http://www.independent.co.uk/news/people/obituary--konrad-zuse-1526795.html> (accessed in April 2012).

6. An online video made at the Deutschen Museum München shows how the Z3 worked, using examples of arithmetical division and square roots: <http://www.youtube.com/>

even though mechanical, using punched tapes.

His main motivation to switch from a mechanical to an electronic mode was a concern about reliability – he wanted to build resilient and fault-tolerant machines – but the Z3 built with electronic relays was logically equivalent to the Z1. The Z1 and Z3 could be programmed and could perform all arithmetical calculations, could load and store information in binary and were capable of floating-point calculations (whereas the Mark I and the ENIAC in the U.S. still represented data in decimals, even though they both operated with binary gates, and were unable to handle floating-point calculations). Zuse decided to use the binary system and metallic plates that could move only in one direction, i.e. they could only shift position, just as modern digital computers do at their lowest working level (Zuse seemed to believe that mechanical devices and digitally based calculations were more reliable as compared to, for example, vacuum tubes, as suggested by Helmut Schreyer, Zuse’s friend.).

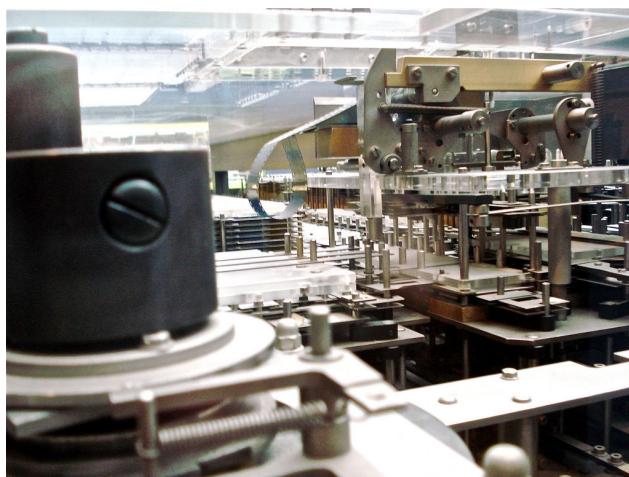


Figure E.1 – Replica of the first mechanical computer designed by Konrad Zuse, the Z1, finished in 1938. It was a binary electrically driven mechanical calculator which used Boolean logic and binary floating point numbers. Picture taken by H. Zenil, *Deutsches Technikmuseum* (“German Museum of Technology”), Berlin.

Zuse and Turing never met but they became acquainted with each other’s work. Zuse mentions Turing’s work in his autobiography, and it is known that Turing was on the program/reviewing committee of at least one colloquium that Zuse attended – but not Turing – at the Max-Planck-Gesellschaft in Göttingen in 1947. Had Turing attended they would actually have met.

[watch?v=J98KVfeC8fU](https://www.youtube.com/watch?v=J98KVfeC8fU) (accessed in April 2012)

But if Zuse didn't hit upon the concept of universal computation, he was interested in another very deep question, the question of the nature of nature: "Is nature digital?" He tended toward an affirmative answer, and his ideas were published, according to Horst Zuse (Konrad's eldest son), in the *Nova Acta Leopoldina*. Horst was born precisely when Konrad was thinking about *Rechnender Raum* for the first time (the common translation into English is *Calculating Space* but the phrase in his native German carries a lot more cognitive weight than its plain English counterpart, in light of the ideas treated in Zuse's piece: calculation, computation of nature, space and/or the universe). H. Zenil (HZ) met Prof. Horst Zuse (of the Technische Universität of Berlin) in the Autumn of 2006 during a conference dinner in Berlin. The conference topic was precisely "Is the Universe a Computer?" (Ist das Universum ein Computer?) and it was held at the Deutschen Technikmuseum and organized to mark the Year of Informatics (Informatik Jahr) in Germany.⁷

Konrad Zuse did, however, acknowledge the problems likely to be faced in attempting to reconcile a digital view of the universe with theories of physics assumed to work in continuum spaces. But according to Konrad Zuse, the laws of physics could be explained in terms of laws of switches or relays (not a surprise as he had experienced the transformation of his machines from mechanical to electronic form through the use of relays), and thought of physical laws as computing approximations captured by mathematical models. It is clear from *Rechnender Raum* that Zuse knew that differential equations could be solved by digital systems and took this fact as evidence in favor of a digital theory.

Years before John von Neumann explained the advantages of a computer architecture in which the processor is separated from the memory, Zuse had already arrived at the same conclusion. As a computer builder in the 1930s, Zuse worked as an amateur completely outside the mathematical community, on his own time, in the evenings and on weekends, in the living room of his parents' house. He did, however, obtain some financial assistance from a local calculating machine manufacturer. He also persuaded Helmut Schreyer, a former university classmate, to work with him. It was on the advice of his friend Schreyer that Zuse moved from mechanical to electro-mechanical, telephonic relays.

In his autobiography⁸, Zuse writes that in 1939, as war broke out, he

7. HZ wrote a blog post about it, available online at <http://www.mathrix.org/liquid/archives/is-the-universe-a-computer>.

8. "The Computer – My Life," published in German by Springer-Verlag in 1993 and translated into English in 2010, the anniversary of Zuse's birth.

was drafted into the infantry to serve on the front lines. He never saw action as a soldier. His military service was to last six months, “six months during which I had plenty of time to contemplate the ideas developed and captured in my diary notes of 1937 and 1938.” He was exempted from active duty and discharged so he could undertake work directly related to weapons development, as a structural engineer in the Special Division F at Henschel Aircraft Company, where remote-controlled flying bombs were developed.

In 1941, shortly after the Z3 was completed, Zuse went back to work as a structural engineer in aircraft construction with Henschel, a day job while starting a company, *Zuse Apparatebau* (Zuse Apparatus Construction), to manufacture his machines. When the Z3 became operational, it was the world’s first practical automatic computer, and for 2 years remained the only one. A second machine, the Z4, was quickly commissioned. During the war Z3 was demonstrated before several departments, yet it was never put into everyday operation. In 1944 the Z3 was destroyed in an air raid but it was reconstructed in 1960 and set up in the *Deutsches Museum* in Munich.

Zuse and Schreyer had, however, to abandon the building where their computer was housed. As the war came to an end, Zuse retreated to Hinterstein, a village in the southeast of Germany, where his eldest son (Horst) was born. There he reconstituted his Z4 computer in a stable, and it became the world’s first operational commercial computer, leased to the ETH Zürich (one of the two universities of the Swiss Federal Institutes of Technology). Then he began working in an area that didn’t require physical resources – computer programming. He devised a language, the Plankalkül (meaning “formal system for planning” or “calculus of programs”; “a universal language” according to Zuse, who compared it to an “artificial brain”), which anticipated some programming concepts that surfaced later, and can be considered the first high-level programming language, although no compiler or interpreter was ever written for it. In 1945, perhaps with the same motivation that led Turing to turn to chess, namely the fact that the game was believed to epitomize human intelligence while seeming highly algorithmic, Zuse worked on chess playing algorithms formulated as routines in his Plankalkül. One year before, in 1944, he had organized his work into a dissertation⁹ which was never defended formally. The title he chose for his work was “Beginnings of a Theory of General Computing,” trying to establish the foundations of what is today generally understood as information processing: “Computing (*Rechnen*)”, he wrote, “means, in general, forming new data from given data according to some rule.” The concept of the algorithm

9. See “The Plankalkül of Konrad Zuse – Revisited” by Friedrich L. Bauer, in “The First Computers – History and Architecture,” cited earlier.

would later replace his concept of *Vorschrift* (or rule). His programming language, like the logic, design and construction of his computing machines, was entirely his own work, carried out in isolation from developments elsewhere.

While still in Hinterstein he wrote a treatise entitled “Freedom and Causality in the Light of the Computing Machine”. In his autobiography he writes: “I think the majority of researchers involved in the development of the computer have at some point in their lives, in one way or another considered the question of the relationship between human free will and causality.” This was to be the major impetus for the work that led to the translation presented in this volume:

“While considering causality it suddenly occurred to me that the universe could be conceived as a gigantic computing machine. I had the relay calculator in mind: relay calculators contain relay chains. When a relay is triggered, the impulse propagates through the entire chain. The thought went through my head that this must also be how a quantum of light propagates. The thought settled firmly; over the years I have developed it into a concept of the *Rechnender Raum*, or ‘computing universe’. However, it was to be another thirty years before I succeeded in formulating the idea correctly.”

In 1967, Zuse suggested that the universe itself was running on a cellular automaton or a similar computational structure, a metaphysical position known today as digital physics, a subject Ed Fredkin had himself taken up before becoming acquainted with the work of Zuse. Excited to discover this work, Fredkin invited Zuse to Cambridge, MA. The translation of *Rechnender Raum* reproduced here, from a German (published) version of Zuse’s ideas, was in fact commissioned during Ed Fredkin’s tenure as Director of MIT’s Project MAC¹⁰ (the AI lab that was a precursor of the current MIT AI labs)

More than twenty years after his *Rechnender Raum*, in Zuse’s autobiography, he wrote:

“In the final analysis, the concept of the computing universe requires a rethinking of ideas, for which physicists are not yet prepared. Yet it is clear that earlier concepts have reached the limits of their possibilities; but no one dares to switch to a fundamentally new track. Yet, with quantization, the preliminary steps

10. Ed Fredkin is also a contributor to *A Computable Universe: Understanding & Exploring Nature as Computation*.



Figure E.2 – (How) Does Nature Compute? A Panel Discussion organized by A. German and H. Zenil during the last day of the 2008 NKS Midwest Conference, featuring (in order): Greg Chaitin , Ed Fredkin, Rob de Ruyter, Anthony Leggett, Cristian Calude, Tommaso Toffoli and Stephen Wolfram, moderated by (from left to right) Gerardo Ortiz, George Johnson and Hector Zenil, at the University of Indiana Bloomington. See <http://www.cs.indiana.edu/~dgerman/2008midwestNKSconference/>

towards a digitalization of physics have already been taken; but only a few physicists have attempted to think along the lines of these new categories of computer science. [...] This was illustrated quite clearly during the conference on the Physics of Computation, held May 6-8, 1981 [at MIT]. What was typical at this conference was that, although the relationship between physics and computer science, and/or computer hardware, was examined in detail, the questions of the physical possibilities and limits of computer hardware still dominated the discussions. The deeper question, to what extent processes in physics can be explained as computer processes, was dealt with only marginally at this otherwise very advanced conference.”

The original of *Rechnender Raum* seems to have been lost. To our knowledge the translation commissioned by Project MAC (the precursor to the current MIT Computer Science and Artificial Intelligence Laboratory or *CSAIL*) was never published in a journal¹¹. It is reproduced here translated into modern $\text{\LaTeX}$, which required quite a bit of work, despite having used OCR

11. Scanned copies of a short German version and the translation into English, accom-

techniques with *Mathematica* first, in order to avoid starting completely from scratch. It is published in this volume without changes, except for perhaps a few corrected typos and redistribution of text and images to fit the book format. The material is at once dated and surprisingly contemporary: “I propose that in an information-theoretic analysis, objects and elementary dimensions of physics must not be complemented by the concept of information, but rather should be explained by it.” Zuse was always aware of the hypothetical nature of his thesis: “The concept of the computing universe is still just a hypothesis; nothing has been proved. However, I am confident that this idea can help unveil the secrets of nature.”

Zuse refers the more skeptical among us to a quote from Freeman Dyson (“Innovation in Physics” published in *Scientific American*, Vol. 199, No. 3, (September 1958), pp. 74-82.): “A few months ago Werner Heisenberg and Wolfgang Pauli believed that they had made an essential step forward in the direction of a theory of elementary particles. Pauli happened to be passing through New York, and was prevailed upon to give a lecture explaining the new ideas to an audience which included Niels Bohr. Pauli spoke for an hour, and then there was a general discussion during which he was criticized rather sharply by the younger generation. Finally Bohr was called on to make a speech summing up the argument. ‘We are all agreed,’ he said, ‘that your theory is crazy. The question which divides us is whether it is crazy enough to have a chance of being correct. My own feeling is that it is not crazy enough.’”

“Imagination,” Zuse used to say, “is the key to all progress.”

A. German and H. Zenil

Bloomington, Indiana, U.S. and Sheffield, U.K.

panied by additional contextual material, are available online at Schmidhuber’s website *Zuse’s thesis* at <http://www.idsia.ch/~juergen/digitalphysics.html>. The German version is also at <http://www.zib.de/zuse/Inhalt/Texte/Chrono/60er/Pdf/76scan.pdf> (links accessed in April 2012)

Bibliographie

- [Aaronson, 2012] Aaronson, S. (2012). Why philosophers should care about computational complexity. In *Computability : Gödel, Turing, Church, and beyond*.
- [Adamatzky, 2015] Adamatzky, A. (2015). *Atlas of Physarum Computing*. World Scientific Publishing Co.
- [Adleman et al., 2001] Adleman, L., Cheng, Q., Goel, A., and Huang, M.-D. (2001). Running time and program size for self-assembled squares. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 740–748. ACM.
- [Adleman, 1999] Adleman, L. M. (1999). Toward a mathematical theory of self-assembly.
- [Adriaans, 2010] Adriaans, P. (2010). A critical analysis of Floridi’s theory of semantic information. *Knowledge, Technology & Policy*, 23(1-2) :41–56.
- [Aggarwal et al., 2005] Aggarwal, G., Cheng, Q., Goldwasser, M. H., Kao, M.-Y., De Espanes, P. M., and Schweller, R. T. (2005). Complexities for generalized models of self-assembly. *SIAM Journal on Computing*, 34(6) :1493–1515.
- [Arkani-Hamed et al., 2010] Arkani-Hamed, N., Cachazo, F., Cheung, C., and Kaplan, J. (2010). The S-matrix in twistor space. *Journal of High Energy Physics*, 2010(3) :1–48.
- [Ashtekar and Lewandowski, 2004] Ashtekar, A. and Lewandowski, J. (2004). Background independent quantum gravity : a status report. *Classical and Quantum Gravity*, 21(15) :R53.
- [Aspect et al., 1982] Aspect, A., Grangier, P., and Roger, G. (1982). Experimental realization of Einstein-Podolsky-Rosen-Bohm Gedankenexperiment : a new violation of Bell’s inequalities. *Physical review letters*, 49(2) :91.
- [Atran and Norenzayan, 2004] Atran, S. and Norenzayan, A. (2004). Religion’s evolutionary landscape : Counterintuition, commitment, compassion, communion. *Behavioral and brain sciences*, 27(06) :713–730.
- [Baddeley, 1992] Baddeley, A. (1992). Working memory. *Science*, 255(5044) :556–559.
- [Bailey, 2000] Bailey, D. (2000). A compendium of BBP-type formulas for mathematical constants. *Preprint*, [http ://crd. lbl. gov/dhbailey/dhbpapers/index. html](http://crd.lbl.gov/dhbailey/dhbpapers/index.html).

- [Bailey et al., 1997] Bailey, D., Borwein, P., and Plouffe, S. (1997). On the rapid computation of various polylogarithmic constants. *Mathematics of Computation of the American Mathematical Society*, 66(218) :903–913.
- [Barrett and Nyhof, 2001] Barrett, J. L. and Nyhof, M. A. (2001). Spreading non-natural concepts : The role of intuitive conceptual structures in memory and transmission of cultural materials. *Journal of Cognition and Culture*, 1(1) :69–100.
- [Barricelli, 1963] Barricelli, N. A. (1963). Numerical testing of evolution theories. *Acta Biotheoretica*, 16(3-4) :99–126.
- [Barrick et al., 2009] Barrick, J. E., Yu, D. S., Yoon, S. H., Jeong, H., Oh, T. K., Schneider, D., Lenski, R. E., and Kim, J. F. (2009). Genome evolution and adaptation in a long-term experiment with *Escherichia coli*. *Nature*, 461(7268) :1243–1247.
- [Barrouillet et al., 2004] Barrouillet, P., Bernardin, S., and Camos, V. (2004). Time constraints and resource sharing in adults’ working memory spans. *Journal of Experimental Psychology : General*, 133(1) :83.
- [Barrouillet et al., 2009] Barrouillet, P., Gavens, N., Vergauwe, E., Gaillard, V., and Camos, V. (2009). Working memory span development : a time-based resource-sharing model account. *Developmental psychology*, 45(2) :477.
- [Bekenstein, 1980] Bekenstein, J. D. (1980). Black-hole thermodynamics. *Physics Today*, 33(1) :24–31.
- [Bekenstein, 2003] Bekenstein, J. D. (2003). Information in the holographic universe. *Scientific American*, 289(2) :58–65.
- [Bell, 1985] Bell, J. S. (1985). Free variables and local causality. *Dialectica*, 39 :103–106.
- [Bennett, 1982] Bennett, C. (1982). The thermodynamics of computation—a review. *International Journal of Theoretical Physics*, 21(12) :905–940.
- [Bennett, 1987] Bennett, C. (1987). Demons, engines and the second law. *Scientific American*, 257(5) :108–116.
- [Bennett, 1988] Bennett, C. (1988). Logical depth and physical complexity. In Herken, R., editor, *The Universal Turing Machine—a Half-Century Survey*, pages 227–257. Oxford University Press.
- [Bennett, 2008] Bennett, C. (2008). A quantum view of the origin of randomness, classicality and complexity.
- [Bennett, 1997] Bennett, K. (1997). *Evolution and ecology : the pace of life*. Cambridge University Press.

- [Blum, 2012] Blum, L. (2012). Alan Turing and the other theory of computation (on Turing’s rounding-off errors in matrix processes) in Barry Cooper and Jan van Leeuwen (eds), *Alan Turing—his work and impact*.
- [Blum, 1971] Blum, M. (1971). On effective procedures for speeding up algorithms. *Journal of the ACM (JACM)*, 18(2) :290–305.
- [Bohr, 1934] Bohr, N. (1934). Proposal for an experimental test of the many-worlds interpretation of quantum mechanics. In *Atomic Theory and the Description of Human Knowledge*, page 19. Cambridge University Press.
- [Bolognesi, 2001] Bolognesi, T. (2001). Reality is ultimately digital, and its program is still undebugged. *Essay Forum FQXi Essay Contest - Is Reality Digital or Analog ?*
- [Borel, 1913] Borel, E. (1913). Mécanique statistique et irréversibilité. *J. Phys.*, 5e série(3) :189—196.
- [Boysen and Hallberg, 2000] Boysen, S. T. and Hallberg, K. I. (2000). Primate numerical competence : contributions toward understanding nonhuman cognition. *Cognitive Science*, 24(3) :423–443.
- [Brenner, 2012] Brenner, S. (2012). Turing centenary : Life’s code script. *Nature*, 482(7386) :461–461.
- [Bridges, 1999] Bridges, D. S. (1999). Can constructive mathematics be applied in physics ? *Journal of Philosophical Logic*, 28(5) :439–453.
- [Bronner, 2010] Bronner, G. (2010). Le succès d’une croyance. *L’Année sociologique*, 60(1) :137–160.
- [Cabello and Joosten, 2011] Cabello, A. and Joosten, J. J. (2011). Hidden variables simulating quantum contextuality increasingly violate the holevo bound. In *Unconventional Computation*, pages 64–76. Springer.
- [Calude, 2002] Calude, C. (2002). Information and randomness : an algorithmic perspective.
- [Calude and Casti, 2004] Calude, C. and Casti, J. (2004). The jumble cruncher. *The New Scientist*, 25 :36–37.
- [Calude and Meyerstein, 1999] Calude, C. and Meyerstein, F. (1999). Is the universe lawful ? *Chaos, Solitons & Fractals*, 10(6) :1075–1084.
- [Calude and Salomaa, 1994] Calude, C. and Salomaa, A. (1994). Algorithmically coding the universe. In Rozenberg, G. and Salomaa, A., editors, *Developments in Language Theory*. World Scientific.
- [Calude and Stay, 2008] Calude, C. and Stay, M. (2008). Most programs stop quickly or never halt. *Advances in Applied Mathematics*, 40(3) :295–308.

- [Calude, 2005] Calude, C. S. (2005). Algorithmic randomness, quantum physics, and incompleteness. In *Machines, computations, and universality*, pages 1–17. Springer.
- [Calude et al., 2010] Calude, C. S., Dinneen, M. J., Dumitrescu, M., and Svozil, K. (2010). Experimental evidence of quantum randomness incomputability. *Phys. Rev. A*, 82 :022102.
- [Calude et al., 2002] Calude, C. S., Dinneen, M. J., Shu, C.-K., et al. (2002). Computing a glimpse of randomness. *Experimental Mathematics*, 11(3) :361–370.
- [Calude et al., 2012] Calude, C. S., Meyerstein, W., and Salomaa, A. (2012). The universe is lawless or “*pantôn chrêmatôn metron anthrôpon einai*”. pages 539–547. World Scientific.
- [Calude and Svozil, 2008] Calude, C. S. and Svozil, K. (2008). Quantum randomness and value indefiniteness. *Advanced Science Letters*, 1(2) :165–168.
- [Casali et al., 2013] Casali, A. G., Gosseries, O., Rosanova, M., Boly, M., Sarasso, S., Casali, K. R., Casarotto, S., Bruno, M.-A., Laureys, S., Tononi, G., et al. (2013). A theoretically based index of consciousness independent of sensory processing and behavior. *Science translational medicine*, 5(198).
- [Catania, 2010] Catania, K. C. (2010). Born knowing : tentacled snakes innately predict future prey behavior. *PloS one*, 5(6) :e10953.
- [Chaitin, 1966] Chaitin, G. (1966). On the length of programs for computing finite binary sequences. *Journal of the ACM (JACM)*, 13(4) :547–569.
- [Chaitin, 1975] Chaitin, G. (1975). A theory of program size formally identical to information theory. *Journal of the Association of Computing Machinery*, 22(3) :329–340.
- [Chaitin, 2001] Chaitin, G. (2001). *Exploring Randomness*. Springer Verlag.
- [Chaitin, 2006] Chaitin, G. (2006). The limits of reason. *Scientific American*, 294(3) :74–81.
- [Chaitin, 2008] Chaitin, G. (2008). *Meta Math! The quest for Omega*. Vintage.
- [Chaitin, 2011] Chaitin, G. (2011). Metaphysics, metamathematics and metabiology. *Randomness Through Computation : Some Answers, More Questions*.
- [Chalmers, 1996] Chalmers, D. J. (1996). Does a rock implement every finite-state automaton? *Synthese*, 108(3) :309–333.

- [Chater, 1999] Chater, N. (1999). The search for simplicity : a fundamental cognitive principle? *The Quarterly Journal of Experimental Psychology : Section A*, 52(2) :273–302.
- [Church and Rosser, 1936] Church, A. and Rosser, J. B. (1936). Some properties of conversion. *Transactions of the American Mathematical Society*, 39(3) :472–482.
- [Clark, 2012] Clark, D. (2012). Work on causality causes Judea Pearl to win prize.
- [Cook, 2004] Cook, M. (2004). Universality in elementary cellular automata. *Complex Systems*, 15(1) :1–40.
- [Cooper, 2009] Cooper, B. (2009). Emergence as a computability-theoretic phenomenon. *Applied Mathematics and Computation*, 215(4) :1351–1360.
- [Cover and Thomas, 2006] Cover, T. and Thomas, J. (2006). *Elements of Information Theory, 2nd edition*. Wiley-Blackwell.
- [Cowan, 2001] Cowan, N. (2001). The magical number 4 in short-term memory : a reconsideration of mental storage capacity. *Behavioral Brain Science*, 24(1) :87–114.
- [Cronin et al., 2006] Cronin, L., Krasnogor, N., Davis, B. G., Alexander, C., Robertson, N., Steinke, J. H., Schroeder, S. L., Khlobystov, A. N., Cooper, G., Gardner, P. M., et al. (2006). The imitation game—a computational chemical approach to recognizing life. *Nature biotechnology*, 24(10) :1203–1206.
- [Daumer et al., 2006] Daumer, M., Duerr, D., Goldstein, S., Maudlin, T., Tumulka, R., and Zanghi, N. (2006). The message of the quantum?
- [de La Mettrie and Thomson, 1996] de La Mettrie, J. O. and Thomson, A. (1996). *Man Machine and Other Writings*. Cambridge Univ Press.
- [de Vladar and Barton, 2011] de Vladar, H. P. and Barton, N. H. (2011). The contribution of statistical physics to evolutionary biology. *Trends in ecology & evolution*, 26(8) :424–432.
- [Dehaene et al., 2006] Dehaene, S., Izard, V., Pica, P., and Spelke, E. (2006). Core knowledge of geometry in an amazonian indigene group. *Science*, 311(5759) :381–384.
- [Delahaye and Zenil, 2007] Delahaye, J. and Zenil, H. (2007). On the Kolmogorov-Chaitin complexity for short sequences. In Calude, C., editor, *Complexity and Randomness : From Leibniz to Chaitin*. World Scientific.
- [Delahaye and Zenil, 2008] Delahaye, J. and Zenil, H. (2008). Towards a stable definition of Kolmogorov-Chaitin complexity. *arXiv preprint arXiv :0804.3459*.

- [Delahaye and Zenil, 2011] Delahaye, J. and Zenil, H. (2011). On the algorithmic nature of the world. pages 477–496.
- [Delahaye and Zenil, 2012] Delahaye, J. and Zenil, H. (2012). Numerical evaluation of algorithmic complexity for short strings : A glance into the innermost structure of randomness. *Applied Mathematics and Computation*, 219(1) :63–77.
- [Delahaye, 2009] Delahaye, J. P. (2009). *Complexité aléatoire et complexité organisée*. Editions Quae.
- [Dennett, 1981] Dennett, D. C. (1981). *Brainstorms : Philosophical essays on mind and psychology*. Number 8. MIT press.
- [Dennett and Mittwoch, 1996] Dennett, D. C. and Mittwoch, U. (1996). Darwin’s dangerous idea : Evolution and the meanings of life. *Annals of Human Genetics*, 60(3) :267–267.
- [Deutsch, 1985] Deutsch, D. (1985). Quantum theory, the Church Turing principle, and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818) :97–117.
- [Deutsch, 2011] Deutsch, D. (2011). *The fabric of reality : the science of parallel universes and its implications*. Penguin UK.
- [DeWitt et al., 1973] DeWitt, B. S., Graham, N., Everett, H., and Wheeler, J. A. (1973). *The many-worlds interpretation of quantum mechanics*, volume 3. Princeton University Press Princeton.
- [Diosi, 1989] Diosi, L. (1989). Models for universal reduction of macroscopic quantum fluctuations. *Physical Review A*, 40(3) :1165.
- [Dodig-Crnkovic, 2007] Dodig-Crnkovic, G. (2007). Where do new ideas come from ? How do they emerge ? Epistemology as computation (information processing). *Cristian S. Calude*, page 263.
- [Dresner, 2010] Dresner, E. (2010). Measurement-theoretic representation and computation-theoretic realization. *The Journal of Philosophy*, 107(6) :275–292.
- [Dubucs, 2006] Dubucs, J. (2006). Unfolding cognitive capacities. In *Reasoning and Cognition : Interdisciplinary Conference on Reasoning Cognition, (Tokyo meeting, December 2005)*, pages 95–101.
- [Dyson, 1998] Dyson, G. (1998). *Darwin among the machines : The evolution of global intelligence*. Da Capo Press.
- [Earman, 1986] Earman, J. (1986). *A primer on determinism*, volume 37. Springer Science & Business Media.

- [Edin et al., 2009] Edin, F., Klingberg, T., Johansson, P., McNab, F., Tegnér, J., and Compte, A. (2009). Mechanism for top-down control of working memory capacity. *Proceedings of the National Academy of Sciences*, 106(16) :6802–6807.
- [Everett III, 1957] Everett III, H. (1957). “Relative state” formulation of quantum mechanics. *Reviews of modern physics*, 29(3) :454.
- [Feynman et al., 2000] Feynman, R., Hey, A., and Pines, D. (2000). *Feynman Lectures on Computation*. Westview Press.
- [Floridi, 2003] Floridi, L. (2003). Information. In Floridi, L., editor, *The Blackwell Guide to Philosophy of Computing and Information*, pages 40–62. Wiley-Blackwell.
- [Floridi, 2004] Floridi, L. (2004). Informational realism. In Weckert, J. and Al-Saggaf, Y., editors, *Computing and Philosophy Conference*, volume 37.
- [Floridi, 2008] Floridi, L. (2008). *Philosophy of computing and information : 5 questions*. Automatic Press/VIP.
- [Fodor, 1975] Fodor, J. A. (1975). *The language of thought*, volume 5. Harvard University Press.
- [Fredkin and Toffoli, 2002] Fredkin, E. and Toffoli, T. (2002). *Conservative logic*. Springer.
- [Gardner, 1970] Gardner, M. (1970). Mathematical games : The fantastic combinations of John Conway’s new solitaire game “Life”. *Scientific American*, 223(4) :120–123.
- [Gauvrit and Morsanyi, 2014] Gauvrit, N. and Morsanyi, K. (2014). The equiprobability bias from a mathematical and psychological perspective. *Advances in Cognitive Psychology*, 10(4) :119.
- [Gauvrit et al., 2014a] Gauvrit, N., Singmann, H., Soler-Toscano, F., and Zenil, H. (2014a). Algorithmic complexity for psychology : A user-friendly implementation of the coding theorem method. *arXiv preprint arXiv :1409.4080*.
- [Gauvrit et al., 2014b] Gauvrit, N., Soler-Toscano, F., and Zenil, H. (2014b). Natural scene statistics mediate the perception of image complexity. *Visual Cognition*, 22(8) :1084–1091.
- [Gauvrit et al., 2011] Gauvrit, N., Zenil, H., and Delahaye, J.-P. (2011). Assessing cognitive randomness : A Kolmogorov complexity approach. *arXiv preprint arXiv :1106.3059*.
- [Gauvrit et al., 2014c] Gauvrit, N., Zenil, H., Delahaye, J.-P., and Soler-Toscano, F. (2014c). Algorithmic complexity for short binary strings applied to psychology : a primer. *Behavior research methods*, 46(3) :732–744.

- [Gödel, 1986] Gödel, K. (1986). On formally undecidable propositions of Principia Mathematica and related systems (1931). In Feferman, S., editor, *Kurt Gödel Collected works*, pages 144–195. Oxford University Press.
- [Godfrey-Smith and Sterelny, 2007] Godfrey-Smith, P. and Sterelny, K. (2007). Biological information.
- [Goodstein, 1944] Goodstein, R. L. (1944). On the restricted ordinal theorem. *The Journal of Symbolic Logic*, 9(02) :33–41.
- [Grafen, 2007] Grafen, A. (2007). The formal Darwinism project : a mid-term report. *Journal of evolutionary biology*, 20(4) :1243–1254.
- [Grafen, 2008] Grafen, A. (2008). The simplest formal argument for fitness optimization. *Journal of genetics*, 87(4) :421–433.
- [Greenberger et al., 1990] Greenberger, D. M., Horne, M. A., Shimony, A., and Zeilinger, A. (1990). Bell’s theorem without inequalities. *Am. J. Phys*, 58(12) :1131–1143.
- [Griffith et al., 2005] Griffith, S., Goldwater, D., and Jacobson, J. M. (2005). Robotics : Self-replication from random parts. *Nature*, 437(7059) :636–636.
- [Grunwald and Vitányi, 2004] Grunwald, P. and Vitányi, P. (2004). Shannon information and Kolmogorov complexity. *arXiv preprint cs/0410002*.
- [Gács, 1974] Gács, P. (1974). Symmetry of algorithmic information. *Doklady Akademii Nauk SSSR*, 218(6) :1265–1267.
- [Hameroff and Penrose, 1996] Hameroff, S. R. and Penrose, R. (1996). Conscious events as orchestrated space-time selections. *Journal of consciousness studies*, 3(1) :36–53.
- [Harnad, 1990] Harnad, S. (1990). The symbol grounding problem. *Physica D : Nonlinear Phenomena*, 42(1) :335–346.
- [Hawking, 1974] Hawking, S. W. (1974). Black hole explosions. *Nature*, 248(5443) :30–31.
- [Hawking and Penrose, 1970] Hawking, S. W. and Penrose, R. (1970). The singularities of gravitational collapse and cosmology. In *Proceedings of the Royal Society of London A : Mathematical, Physical and Engineering Sciences*, volume 314, pages 529–548. The Royal Society.
- [Helm and Young, 1971] Helm, J. and Young, P. (1971). On size vs. efficiency for programs admitting speed-ups. *The Journal of Symbolic Logic*, 36(01) :21–27.
- [Hodgkin and Huxley, 1952] Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4) :500–544.

- [Hofstadter, 2008] Hofstadter, D. R. (2008). *I am a strange loop*. Basic books.
- [Hopfield, 1994] Hopfield, J. (1994). Physics, computation, and why biology looks so different. *Journal of Theoretical Biology*, 171(1) :53–60.
- [Hsu et al., 2010] Hsu, A. S., Griffiths, T. L., and Schreiber, E. (2010). Subjective randomness and natural scene statistics. *Psychonomic bulletin & review*, 17(5) :624–629.
- [Humphreys, 1996] Humphreys, P. (1996). Aspects of emergence. *Philosophical Topics*, 24(1) :53–70.
- [Humphreys, 1997] Humphreys, P. (1997). How properties emerge. *Philosophy of Science*, pages 1–17.
- [Imbert, 2008] Imbert, C. (2008). *L’opacité intrinsèque de la nature : théories connues, phénomènes difficiles à expliquer et limites de la science*. PhD thesis, Paris 1.
- [Israeli and Goldenfeld, 2004] Israeli, N. and Goldenfeld, N. (2004). Computational irreducibility and the predictability of complex physical systems. *Physical review letters*, 92(7).
- [Israeli and Goldenfeld, 2006] Israeli, N. and Goldenfeld, N. (2006). Coarse-graining of cellular automata, emergence, and the predictability of complex systems. *Physical Review E*, 73(2) :026203.
- [Johnston et al., 2011] Johnston, I. G., Ahnert, S. E., Doye, J. P., and Louis, A. A. (2011). Evolutionary dynamics in a simple model of self-assembly. *Physical Review E*, 83(6) :066105.
- [Joosten et al., 2011] Joosten, J., Soler-Toscano, F., and Zenil, H. (2011). Program-size versus time complexity, speed-up and slowdown phenomena in small Turing machines. *International Journal of Unconventional Computing*, Special Issue on Physics and Computation 2011.
- [Kauffman, 1993] Kauffman, S. A. (1993). *The origins of order : Self-organization and selection in evolution*. Oxford university press.
- [Kempe et al., 2015] Kempe, V., Gauvrit, N., and Forsyth, D. (2015). Structure emerges faster during cultural transmission in children than in adults. *Cognition*, 136 :247–254.
- [Kersten and Earles, 2001] Kersten, A. W. and Earles, J. L. (2001). Less really is more for adults learning a miniature artificial language. *Journal of Memory and Language*, 44(2) :250–273.
- [Kirby and Paris, 1982] Kirby, L. and Paris, J. (1982). Accessible independence results for Peano arithmetic. In *Bulletin of the London Mathematical Society*. Citeseer.

- [Kirchherr et al., 1997] Kirchherr, W., Li, M., and Vitányi, P. (1997). The miraculous universal distribution. *The Mathematical Intelligencer*, 19(4) :7–15.
- [Kochen and Specker, 1975] Kochen, S. and Specker, E. (1975). The problem of hidden variables in quantum mechanics. In *The Logico-Algebraic Approach to Quantum Mechanics*, pages 293–328. Springer.
- [Kolmogorov, 1965] Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems of Information and Transmission*, 1(1) :1–7.
- [Kronheimer and Penrose, 1967] Kronheimer, E. and Penrose, R. (1967). On the structure of causal spaces. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 63, pages 481–501. Cambridge Univ Press.
- [Kryazhimskiy et al., 2014] Kryazhimskiy, S., Rice, D. P., Jerison, E. R., and Desai, M. M. (2014). Global epistasis makes adaptation predictable despite sequence-level stochasticity. *Science*, 344(6191) :1519–1522.
- [Landauer, 1961] Landauer, R. (1961). Irreversibility and heat generation in the computing process. *IBM journal of research and development*, 5(3) :183–191.
- [Landauer, 1991] Landauer, R. (1991). Information is physical. *Physics Today*, pages 23–29.
- [Landauer, 1996] Landauer, R. (1996). The physical nature of information. *Physics letters A*, 217(4) :188–193.
- [Langton, 1986] Langton, C. G. (1986). Studying artificial life with cellular automata. *Physica D : Nonlinear Phenomena*, 22(1) :120–149.
- [Lecoutre, 1992] Lecoutre, M.-P. (1992). Cognitive models and problem spaces in “purely random” situations. *Educational studies in mathematics*, 23(6) :557–568.
- [Leff and Rex, 2002] Leff, H. and Rex, A. F. (2002). *Maxwell’s Demon 2 Entropy, Classical and Quantum Information, Computing*. CRC Press.
- [Lenski et al., 2003] Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423(6936) :139–144.
- [Levin, 1974] Levin, L. (1974). Laws of information conservation (non-growth) and aspects of the foundation of probability theory. *Problemy Peredachi Informatsii*, 10(3) :30–35.
- [Levin, 1977] Levin, L. (1977). On a concrete method of assigning complexity measures. *Doklady Akademii nauk SSSR*, 18(3) :727–731.

- [Levin, 2003] Levin, L. (2003). The tale of one-way functions. *Problems of Information Transmission*, 39(1) :92–103.
- [Li et al., 1996] Li, H., Helling, R., Tang, C., and Wingreen, N. (1996). Emergence of preferred structures in a simple model of protein folding. *Science*, 273(5275) :666–669.
- [Li et al., 2001] Li, M., Badger, J. H., Chen, X., Kwong, S., Kearney, P., and Zhang, H. (2001). An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, 17(2) :149–154.
- [Li et al., 2004] Li, M., Chen, X., Li, X., Ma, B., and Vitányi, P. M. (2004). The similarity metric. *Information Theory, IEEE Transactions on*, 50(12) :3250–3264.
- [Li and Vitányi, 2008] Li, M. and Vitányi, P. (2008). *An Introduction to Kolmogorov Complexity and Its Applications*. Springer.
- [Li, 1999] Li, Z. (1999). Algebraic properties of DNA operations. *BioSystems*, 52(1-3).
- [Lin and Rado, 1965] Lin, S. and Rado, T. (1965). Computer studies of Turing machine problems. *Journal of the ACM (JACM)*, 12(2) :196–212.
- [Livnat and Pippenger, 2006] Livnat, A. and Pippenger, N. (2006). An optimal brain can be composed of conflicting agents. *Proceedings of the National Academy of Sciences of the United States of America*, 103(9) :3198–3202.
- [Lloyd, 2002] Lloyd, S. (2002). Computational capacity of the universe. *Physical Review Letters*, 88(23) :237901.
- [Lloyd, 2006] Lloyd, S. (2006). *Programming the universe : a quantum computer scientist takes on the cosmos*. Random House LLC.
- [Lorenz, 1963] Lorenz, E. (1963). Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2) :130–141.
- [Ma and Xu, 2013] Ma, L. and Xu, F. (2013). Preverbal infants infer intentional agents from the perception of regularity. *Developmental psychology*, 49(7) :1330.
- [Maguire et al.,] Maguire, P., Moser, P., Maguire, R., and Griffith, V. Is consciousness computable? quantifying integrated information using algorithmic information theory. In *Proceedings of the 36th Annual Conference of the Cognitive Science Society*. Cognitive Science Society.
- [Mandler and Shebo, 1982] Mandler, G. and Shebo, B. J. (1982). Subitizing : an analysis of its component processes. *Journal of Experimental Psychology : General*, 111(1) :1.

- [Margolus, 1984] Margolus, N. (1984). Physics-like models of computation. *Physica D : Nonlinear Phenomena*, 10(1) :81–95.
- [Marshall et al., 2003] Marshall, W., Simon, C., Penrose, R., and Bouwmeester, D. (2003). Towards quantum superpositions of a mirror. *Physical Review Letters*, 91(13) :130401.
- [Martin-Lof, 1966] Martin-Lof, P. (1966). The definition of random sequences. *Information and Control*, 9(6) :602–619.
- [Mathy et al., 2014] Mathy, F., Chekaf, M., and Gauvrit, N. (2014). Chunking on the fly in working memory and its relationship to intelligence. In *Abstracts of the 55th Annual meeting of the Psychonomic Society (Abstract #148)*, page 32.
- [Mathy and Feldman, 2012] Mathy, F. and Feldman, J. (2012). What’s magic about magic numbers? chunking and data compression in short-term memory. *Cognition*, 122(3) :346–362.
- [Matthews, 2013] Matthews, W. J. (2013). Relatively random : Context effects on perceived randomness and predicted outcomes. *Journal of Experimental Psychology : Learning, Memory, and Cognition*, 39(5) :1642.
- [McAllister, 2003] McAllister, J. (2003). Algorithmic randomness in empirical data. *Studies in History and Philosophy of Science Part A*, 34(3) :633–646.
- [McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4) :115–133.
- [McGinnis et al., 1983] McGinnis, W., Levine, M., Hafen, E., Kuroiwa, A., and Gehring, W. (1983). A conserved DNA sequence in homoeotic genes of the *Drosophila* Antennapedia and bithorax complexes. *Nature*, 308(5958) :428–433.
- [McNamara and Dall, 2010] McNamara, J. M. and Dall, S. R. (2010). Information is a fitness enhancing resource. *Oikos*, 119(2) :231–236.
- [Meyer and Fischer, 1972] Meyer, A. and Fischer, P. (1972). Computational speed-up by effective operators. *The Journal of Symbolic Logic*, 37(01) :55–68.
- [Miller, 1956] Miller, G. A. (1956). The magical number seven, plus or minus two : some limits on our capacity for processing information. *Psychological review*, 63(2) :81.
- [Minsky, 1967] Minsky, M. L. (1967). *Computation : finite and infinite machines*. Prentice-Hall, Inc.

- [Mitchell, 2011] Mitchell, M. (2011). Ubiquity symposium : biological computation. *Ubiquity*, 2011(February) :3.
- [Morange, 2012] Morange, M. (2012). *Les secrets du vivant*. La Découverte.
- [Nicolis and Prigogine, 1989] Nicolis, G. and Prigogine, I. (1989). *Exploring complexity : An introduction*. W. H. Freeman.
- [Oberauer et al., 2004] Oberauer, K., Lange, E., and Engle, R. W. (2004). Working memory capacity and resistance to interference. *Journal of Memory and Language*, 51(1) :80–96.
- [Oizumi et al., 2014] Oizumi, M., Albantakis, L., and Tononi, G. (2014). From the phenomenology to the mechanisms of consciousness : integrated information theory 3.0. *PLoS computational biology*, 10(5) :e1003588.
- [Parrochia, 1996] Parrochia, D. (1996). L’idée d’une information-limite. *Philosophie du langage et informatique, Hermes*.
- [Parsons, 2012] Parsons, W. T. (2012). Patterns in the noise : Physics as the ultimate environmental science. *FQXi’s, Essay Contest, Questioning the Foundations*.
- [Paz Flanagan et al., 2011] Paz Flanagan, T., Letendre, K., Burnside, W., Fricke, G. M., and Moses, M. (2011). How ants turn information into food. In *Artificial Life (ALIFE), 2011 IEEE Symposium on*, pages 178–185. IEEE.
- [Peng et al., 2014] Peng, Z., Genewein, T., and Braun, D. A. (2014). Assessing randomness and complexity in human motion trajectories through analysis of symbolic sequences. *Frontiers in human neuroscience*, 8.
- [Penrose, 1965] Penrose, R. (1965). Gravitational collapse and space-time singularities. *Physical Review Letters*, 14(3) :57–59.
- [Penrose, 1967] Penrose, R. (1967). Twistor algebra. *Journal of Mathematical physics*, 8(2) :345–366.
- [Penrose, 1971] Penrose, R. (1971). Angular momentum : an approach to combinatorial space-time. *Quantum theory and beyond*, pages 151–180.
- [Penrose, 1994] Penrose, R. (1994). Shadows of the mind : an approach to the missing science of consciousness.
- [Penrose, 1996] Penrose, R. (1996). On gravity’s role in quantum state reduction. *General relativity and gravitation*, 28(5) :581–600.
- [Penrose, 1998] Penrose, R. (1998). Can a computer understand? *Essays on the new sciences of the mind*. Princeton, Princeton UP.
- [Penrose, 1999] Penrose, R. (1999). *The emperor’s new mind : concerning computers, minds, and the laws of physics*. Oxford University Press.

- [Penrose, 2005] Penrose, R. (2005). The twistor approach to space-time structures. *100 years of relativity*, 1 :465–505.
- [Penrose, 2009] Penrose, R. (2009). Black holes, quantum theory and cosmology. In *Journal of Physics : Conference Series*, volume 174, page 012001. IOP Publishing.
- [Penrose, 2012] Penrose, R. (2012). On attempting to model the mathematical mind. In Cooper, B. and Hodges, A., editors, *The Alan Turing Year - The Once and Future Turing*. Cambridge University Press.
- [Penrose, 2014] Penrose, R. (2014). *The Road to Reality : A Complete Guide to the Laws of the Universe*. Knopf.
- [Penrose and Hameroff, 2011] Penrose, R. and Hameroff, S. (2011). Consciousness in the universe : Neuroscience, quantum space-time geometry and Orch OR theory. *Journal of Cosmology*, 14 :1–17.
- [Pepperberg, 2006] Pepperberg, I. M. (2006). Grey parrot numerical competence : a review. *Animal cognition*, 9(4) :377–391.
- [Piccinini, 2008] Piccinini, G. (2008). Computers. *Pacific Philosophical Quarterly*, 89(1) :32–73.
- [Popper, 1988] Popper, K. (1988). *The open universe : An argument for indeterminism*, volume 2. Psychology Press.
- [Porter, 2012] Porter, C. P. (2012). *Mathematical and philosophical perspectives on algorithmic randomness*. PhD thesis, University of Notre Dame.
- [Pour-El and Richards, 1981] Pour-El, M. B. and Richards, J. I. (1981). The wave equation with computable initial data such that its unique solution is not computable. *Advances in Mathematics*, 39(3) :215–239.
- [Pour-El and Richards, 1988] Pour-El, M. B. and Richards, J. I. (1988). Computability in analysis and physics. *AMC*, 10 :12.
- [Putnam, 1988] Putnam, H. (1988). *Representation and reality*, volume 454. Cambridge University Press.
- [Rado, 1962] Rado, T. (1962). On noncomputable functions. *Bell System Technical Journal*, 41(3) :877–884.
- [Ratcliff et al., 2012] Ratcliff, W. C., Denison, R. F., Borrello, M., and Travisano, M. (2012). Experimental evolution of multicellularity. *Proceedings of the National Academy of Sciences*, 109(5) :1595–1600.
- [Reznikova et al., 2012] Reznikova, Z., Panteleeva, S., and Danzanov, Z. (2012). A new method for evaluating the complexity of animal behavioral patterns based on the notion of Kolmogorov complexity, with ants’ hunting behavior as an example. *Neurocomputing*, 84 :58–64.

- [Reznikova and Ryabko, 2011] Reznikova, Z. and Ryabko, B. (2011). Numerical competence in animals, with an insight from ants. *Behaviour*, 148(4) :405–434.
- [Reznikova and Ryabko, 2012] Reznikova, Z. and Ryabko, B. (2012). Ants and bits. *IEEE Information Theory Society Newsletter*, 62(5) :17–20.
- [Roederer, 2006] Roederer, J. G. (2006). *Information and its Role in Nature*. Springer Science & Business Media.
- [Ronald et al., 1999] Ronald, E. M., Sipper, M., and Capcarrère, M. S. (1999). Design, observation, surprise! A test of emergence. *Artificial Life*, 5(3) :225–239.
- [Rothmund et al., 2004] Rothmund, P. W., Papadakis, N., and Winfree, E. (2004). Algorithmic self-assembly of DNA sierpinski triangles. *PLoS biology*, 2(12) :e424.
- [Rothmund and Winfree, 2000] Rothmund, P. W. and Winfree, E. (2000). The program-size complexity of self-assembled squares. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 459–468. ACM.
- [Russell, 1927] Russell, B. (1927). The analysis of matter. *International library of psychology, philosophy and scientific method*.
- [Ryabko and Reznikova, 2009] Ryabko, B. and Reznikova, Z. (2009). The use of ideas of information theory for studying “language” and intelligence in ants. *Entropy*, 11(4) :836–853.
- [Sahu et al.,] Sahu, S., Ghosh, S., Hirata, K., Fujita, D., and Bandyopadhyay, A. Ultrafast microtubule growth through radio-frequency-induced resonant excitation of tubulin and small-molecule drugs. *Nature Materials*.
- [Sakharov, 1991] Sakharov, A. D. (1991). Vacuum quantum fluctuations in curved space and the theory of gravitation. *Physics-Uspekhi*, 34(5) :394–394.
- [Sato et al., 2001] Sato, H., Yoshioka, T., Konagaya, A., and Toyoda, T. (2001). DNA data compression in the post genome era. *Genome Informatics*, 12 :512–514.
- [Schnorr, 1972] Schnorr, C. (1972). Process complexity and effective random tests. In *Proceedings of the fourth annual ACM symposium on Theory of computing*, pages 168–176. ACM.
- [Schrödinger, 1951] Schrödinger, E. (1951). *Science and humanism : Physics in our time*. University Press.

- [Scott, 1970] Scott, D. (1970). *Outline of a mathematical theory of computation*. Oxford University Computing Laboratory, Programming Research Group.
- [Scott and Weiner, 1984] Scott, M. P. and Weiner, A. J. (1984). Structural relationships among genes that control development : sequence homology between the Antennapedia, Ultrabithorax, and fushi tarazu loci of *Drosophila*. *Proceedings of the National Academy of Sciences*, 81(13) :4115–4119.
- [Searle, 1980] Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and brain sciences*, 3(03) :417–424.
- [Searle, 1990] Searle, J. R. (1990). Is the brain a digital computer? In *Proceedings and addresses of the american philosophical association*, pages 21–37. JSTOR.
- [Searle, 1992] Searle, J. R. (1992). *The rediscovery of the mind*. MIT press.
- [Sethna, 2006] Sethna, J. (2006). *Statistical mechanics : entropy, order parameters, and complexity*, volume 14. Oxford University Press.
- [Shannon, 1948] Shannon, C. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27 :379–423 and 623–656.
- [Shetty et al., 2008] Shetty, R. P., Endy, D., and Knight Jr, T. F. (2008). Engineering Biobrick vectors from Biobrick parts. *Journal of biological engineering*, 2(1) :1–12.
- [Smith and Wonnacott, 2010] Smith, K. and Wonnacott, E. (2010). Eliminating unpredictable variation through iterated learning. *Cognition*, 116(3) :444–449.
- [Soler-Toscano et al., 2014] Soler-Toscano, F., Zenil, H., Delahaye, J.-P., and Gauvrit, N. (2014). Calculating kolmogorov complexity from the output frequency distributions of small Turing machines. *PloS one*, 9(5) :e96223.
- [Solomonoff, 1960] Solomonoff, R. (1960). A preliminary report on a general theory of inductive inference (revision of report v-131). Technical report, United States Air Force, Office of Scientific Research.
- [Solomonoff, 1964] Solomonoff, R. (1964). A formal theory of inductive inference : Parts i and ii. *Information and Control*, 7(1 and 2) :1–22 and 224–254.
- [Solovay,] Solovay, R. A version of omega for which zfc cannot predict a single bit. In *In Finite Versus Infinite. Contributions to an Eternal Dilemma* (eds. C. Calude and G. Păun), pages 323–334. Springer-Verlag.
- [Spelke and Kinzler, 2007] Spelke, E. S. and Kinzler, K. D. (2007). Core knowledge. *Developmental science*, 10(1) :89–96.

- [Stannett, 2003] Stannett, M. (2003). Computation and hypercomputation. *Minds and Machines*, 13(1) :115–153.
- [Sterkenburg, 2013] Sterkenburg, T. F. (2013). The foundations of Solomonoff prediction. Master’s thesis, Universiteit Utrecht, the Netherlands.
- [Susskind, 1995] Susskind, L. (1995). The world as a hologram. *Journal of Mathematical Physics*, 36(11) :6377–6396.
- [Sutner, 2003] Sutner, K. (2003). Cellular automata and intermediate degrees. *Theoretical Computer Science*, 296(2) :365–375.
- [Sutner, 2011] Sutner, K. (2011). Computational processes, observers and Turing incompleteness. *Theoretical Computer Science*, 412(1) :183–190.
- [Svozil, 1993] Svozil, K. (1993). *Randomness & undecidability in physics*. World Scientific.
- [Szilard, 1929] Szilard, L. (1929). Über die Entropieverminderung in einem thermodynamischen System bei Eingriffen intelligenter Wesen. *Zeitschrift für Physik*, 53(11-12) :840–856.
- [Téglás et al., 2011] Téglás, E., Vul, E., Girotto, V., Gonzalez, M., Tenenbaum, J. B., and Bonatti, L. L. (2011). Pure reasoning in 12-month-old infants as probabilistic inference. *science*, 332(6033) :1054–1059.
- [Tegmark, 2000] Tegmark, M. (2000). Importance of quantum decoherence in brain processes. *Physical Review E*, 61(4) :4194.
- [Tegmark, 2007] Tegmark, M. (2007). The multiverse hierarchy. In Carr, B., editor, *Universe or Multiverse ?* Cambridge University Press.
- [t’Hooft, 2001] t’Hooft, G. (2001). The holographic principle. In *Basics and highlights in fundamental physics : proceedings of the International School of Subnuclear Physics*, volume 37, pages 72–100. World Scientific.
- [Turing, 1936] Turing, A. (1936). On computable numbers, with an application to the Entscheidungsproblem. *J. of Math*, 58 :345–363.
- [Turing, 1937] Turing, A. (1937). On computable numbers, with an application to the entscheidungsproblem : A correction. In *Proceedings of the London Mathematical Society*, volume 43, pages 544–546.
- [Turing, 1939] Turing, A. (1939). Systems of logic based on ordinals. *Proceedings of the London Mathematical Society*, 2(1) :161–228.
- [Turing, 1948] Turing, A. (1948). Intelligent machinery, a heretical theory.
- [Turing, 1950] Turing, A. (1950). Computing machinery and intelligence. *Mind*, pages 433–460.
- [Tversky and Kahneman, 1974] Tversky, A. and Kahneman, D. (1974). Judgment under uncertainty : Heuristics and biases. *science*, 185(4157) :1124–1131.

- [Twardy and Gardner, 2005] Twardy, C. and Gardner, S. (2005). Empirical data sets are algorithmically compressible : Reply to McAllister. *Studies in History and Philosophy of Science Part A*, 36(2) :391–402.
- [Vallinder, 2012] Vallinder, A. (2012). Solomonoff induction : A solution to the problem of the priors ? Master’s thesis, Lund University, Sweden.
- [van Emde Boas, 1975] van Emde Boas, P. (1975). Ten years of speedup. In *Mathematical Foundations of Computer Science 1975 4th Symposium, Mariánské Lázně, September 1–5, 1975*, pages 13–29. Springer.
- [Von Mises, 1928] Von Mises, R. (1928). Wahrscheinlichkeit statistik und wahrheit.
- [Von Neumann, 2013] Von Neumann, J. (2013). *Mathematische grundlagen der quantenmechanik*, volume 38. Springer-Verlag.
- [Wagner, 2013] Wagner, A. (2013). *Robustness and evolvability in living systems*. Princeton University Press.
- [Wang, 1957] Wang, H. (1957). A variant to turing’s theory of computing machines. *Journal of the Association for Computing Machinery*, 4 :23–25.
- [Wang et al., 2014] Wang, Z., Li, Y., Childress, A. R., and Detre, J. A. (2014). Brain entropy mapping using fmri. *PloS one*, 9(3) :e89948.
- [Weihrauch, 2000] Weihrauch, K. (2000). *Computable analysis : an introduction*. Springer Science & Business Media.
- [Weihs et al., 1998] Weihs, G., Jennewein, T., Simon, C., Weinfurter, H., and Zeilinger, A. (1998). Violation of bell’s inequality under strict einstein locality conditions. *Physical Review Letters*, 81(23) :5039.
- [Weinberg, 2010] Weinberg, S. (2010). Dr. unification. interview by Amir D. Aczel. *Scientific American*, 303(5) :64–67.
- [Wheeler, 1990] Wheeler, J. (1990). Information, physics, quantum : The search for links. In Zurek, W., editor, *Complexity, Entropy, and the Physics of Information*. Addison-Wesley.
- [Wheeler, 1962] Wheeler, J. A. (1962). Geometrodynamics.
- [Wheeler, 1983] Wheeler, J. A. (1983). Law without law. *Quantum theory and measurement*, pages 182–213.
- [Winfree, 1998a] Winfree, E. (1998a). *Algorithmic self-assembly of DNA*. PhD thesis, California Institute of Technology.
- [Winfree, 1998b] Winfree, E. (1998b). Simulations of computing by self-assembly.
- [Wolfram, 1984] Wolfram, S. (1984). Computation theory of cellular automata. *Communications in Mathematical Physics*, 96(1) :15–57.

- [Wolfram, 2002] Wolfram, S. (2002). *A New Kind of Science*. Wolfram Media.
- [Woods and Neary, 2009] Woods, D. and Neary, T. (2009). The complexity of small universal Turing machines : A survey. *Theoretical Computer Science*, 410(4) :443–450.
- [Xu and Garcia, 2008] Xu, F. and Garcia, V. (2008). Intuitive statistics by 8-month-old infants. *Proceedings of the National Academy of Sciences*, 105(13) :5012–5015.
- [Xu et al., 2005] Xu, F., Spelke, E. S., and Goddard, S. (2005). Number sense in human infants. *Developmental science*, 8(1) :88–101.
- [Young and Corey, 1990] Young, D. A. and Corey, E. M. (1990). Lattice models of biological growth. *Physical Review A*, 41(12) :7024.
- [Zeilinger, 2005] Zeilinger, A. (2005). The message of the quantum. *Nature*, 438(7069) :743–743.
- [Zenil, 2010] Zenil, H. (2010). Compression-based investigation of the dynamical properties of cellular automata and other systems. *Complex Systems*.
- [Zenil, 2011] Zenil, H. (2011). The world is either algorithmic or mostly random. *FXQi Contest : Is Reality Digital or Analog ?* 3rd Prize Winner.
- [Zenil, 2012a] Zenil, H. (2012a). *A Computable Universe. Understanding and Exploring Nature as Computation*. World Scientific Publishing Company/Imperial College Press.
- [Zenil, 2012b] Zenil, H. (2012b). Information theory and computational thermodynamics : Lessons for biology from physics. *Information*, 3(4) :739–750.
- [Zenil, 2012c] Zenil, H. (2012c). *Irreducibility and Computational Equivalence : 10 Years After Wolfram’s A New Kind of Science*, volume 2. Springer Science & Business Media.
- [Zenil, 2012d] Zenil, H. (2012d). On the dynamic qualitative behaviour of universal computation. *Complex Systems*, 20(3).
- [Zenil, 2012e] Zenil, H. (2012e). Programmability for natural computation and the game of life as a case study. *Journal of Experimental and Theoretical Artificial Intelligence*.
- [Zenil, 2013a] Zenil, H. (2013a). Algorithmic complexity of animal behaviour : From communication to cognition. In *Theory and Practice of Natural Computing Second International Conference Proceedings, TPNC 2013. Cáceres, Spain, December 3-5*.
- [Zenil, 2013b] Zenil, H. (2013b). *Une approche expérimentale la théorie algorithmique de la complexité*. PhD thesis, Université de Paris.

- [Zenil, 2014a] Zenil, H. (2014a). Algorithmicity and programmability in natural computing with the Game of Life as in silico case study. *Journal of Experimental & Theoretical Artificial Intelligence*, (ahead-of-print) :1–13.
- [Zenil, 2014b] Zenil, H. (2014b). What is nature-like computation ? A behavioural approach and a notion of programmability. *Philosophy & Technology*, 27(3) :399–421.
- [Zenil, 2015] Zenil, H. ((submitted)2015). Numerical estimations of algorithmic complexity and logical depth with an application to animal behaviour.
- [Zenil, pear] Zenil, H. ((to appear)). Quantifying natural and artificial intelligence in robots and natural systems with an algorithmic behavioural test. In Bonsignorio, F., del Pobil, A., Messina, E., and Hallam, J., editors, *Metrics of sensory motor integration in robots and animals*. Springer.
- [Zenil et al., 2012a] Zenil, H., Delahaye, J.-P., and Gaucherel, C. (2012a). Image characterization and classification by physical complexity. *Complexity*, 17(3) :26–42.
- [Zenil et al., 2012b] Zenil, H., Gershenson, C., Marshall, J. A., and Rosenblueth, D. A. (2012b). Life as thermodynamic evidence of algorithmic structure in natural environments. *Entropy*, 14(11) :2173–2191.
- [Zenil and Hernandez-Quiroz, 2007] Zenil, H. and Hernandez-Quiroz, F. (2007). On the possible computational power of the human mind. *World-views, science and us : Philosophy and complexity : World Scientific*, pages 315–334.
- [Zenil and Marshall, 2013] Zenil, H. and Marshall, J. (2013). Some aspects of computation essential to evolution and life. *Ubiquity*, pages 1–16.
- [Zenil et al., 2014] Zenil, H., Soler-Toscano, F., Dingle, K., and Louis, A. A. (2014). Correlation of automorphism group size and topological properties with program-size complexity evaluations of graphs and complex networks. *Physica A : Statistical Mechanics and Its Applications*, 404 :341–358.
- [Zenil et al., 2012c] Zenil, H., Soler-Toscano, F., and Joosten, J. J. (2012c). Empirical encounters with computational irreducibility and unpredictability. *Minds and Machines*, 22(3) :149–165.
- [Zenil and Villarreal-Zapata, 2013] Zenil, H. and Villarreal-Zapata, E. (2013). Asymptotic behaviour and ratios of complexity in cellular automata. *International Journal of Bifurcation and Chaos*, 13(9).
- [Zuse, 1969] Zuse, K. (1969). *Rechnender Raum (Calculating Space)*.
- [Zvonkin and Levin., 1970] Zvonkin, A. and Levin., L. (1970). The complexity of finite objects and the algorithmic concepts of information and randomness. *Russian Math. Surveys*.

- [Zwirn, 2013] Zwirn, H. (2013). Unpredictability and computational irreducibility. In *In Zenil, H. (ed) Irreducibility and Computational Equivalence ECC vol 2*, pages 273–295. Springer-Verlag.
- [Zwirn, 2015] Zwirn, H. ((forthcoming)2015). Computational irreducibility and computational analogy. *Complex Systems*.
- [Zwirn, 2000] Zwirn, H. P. (2000). *Limites de la connaissance (Les)*. Odile Jacob.

Index

- λ calculus, 513
- Aaronson, S., 265
- Adamatzky, A., 227
- Adams, D., 415
- analog computers, 459
- analog instruments, 460
- automaton theory, 456
- Babbage, C., 433
- Backus, J., 418
- Bailey-Borwein-Plouffe, 52
- Barricelli, N., 163, 176
- Bauer, A., 426, 427
- Bekenstein, J., 150
- Bell's experiment, 409
- Bell's inequality, 409
- Ben Jacob, E., 392
- Bennett, C., iii, v, xxviii, 7, 49, 50, 53, 108, 118, 119, 140, 151–153, 201, 202, 209, 214–216, 392, 411
- Berry, G., 30
- billiard ball computer, 153
- binary places, 500
- Blum, M., 68
- Bohr, N., 141, 520
- Bolognesi, T., 109, 150, 153, 391, 394, 396, 398, 422, 423
- Boltue, P., 218
- Boltzmann, L., 7
- Boolos, G., 30
- Borel, E., 37
- Boyle, R., 63
- Brenner, S., 261
- Bringsjord, S., 218
- Cabello, A., 91, 109, 111, 407–411, 428
- calculabilité, 3
- Calude, C., xxvii, 90, 108, 109, 111, 131, 133, 167, 191, 210, 324, 325, 330, 391, 393, 401–404, 406–409
- Cantor, G., 18, 424
- Capcarrère, M., 191
- Cartesian space, 492
- Casti, J., 329
- causality, 502
- cellular automata, 57, 457, 491
- cellular automaton, 458, 459
- Chaitin, G., xxviii, 3, 6, 7, 9, 10, 14, 19, 20, 24, 25, 30–32, 44–48, 90, 108, 131, 140, 167, 195, 213–216, 293, 295, 297, 299–302, 304–307, 310, 312–318,

- 321, 323,
325–334,
403, 433, 519
- Chalmers, D., 253
- Church, A., 19, 21,
261,
417–419,
424, 433, 513
- cognition, 259
- computation
réversible,
153
- Cook, M., 54
- Cooper, B., 109,
178, 179, 191
- cybernetics, 511
- Démon de Maxwell,
185
- Dall S., 184
- Darwin, C.,
161–163,
175, 176,
187, 197
- Davies, P., 293, 325,
401
- Davis, M., 417, 418
- De Mol, L., 416–419
- Delahaye, J., 42
- Delahaye, J.-P., 217
- Dennett, D., 195,
248
- Descartes, R., 247
- determinism, 504
- Deutsch, D., xxviii,
69, 91, 108,
109, 146,
316, 325,
331, 402,
411, 434, 438
- differential
equations,
464
- digital computer,
461
- digital computers,
459
- digital particle, 473
- digitalization, 467
- Doria, F., 400, 414
- Dubucs, J., 52
- Dyson, F., 520
- Dyson, G., 163, 176
- Einstein, A., 99,
101, 102,
141, 143, 432
- Feynman, R., iii, v,
89, 90, 110,
140, 148,
150, 216, 411
- flip-flops, 496
- Fredkin, E., iii, v,
xxviii,
90–92, 108,
110, 150,
152, 153,
170, 216,
391, 396,
397, 399,
433, 518, 519
- Gödel, K., 19, 30,
64, 133, 264,
317, 329,
330, 433,
445–448, 450
- Gács, P., 215
- general relativity,
466
- German, A., 108,
453
- Goldenfeld, N., 65
- gravitation, 467
- grid points, 507
- Grover, L., xxviii
- Harhard, S., 248
- Hawking, S., 107
- Hewitt, C., 422
- Hilbert, D., 416,
424, 425, 513
- Hobbes, T., 191
- Hofstadter, D., 263
- Hooke, R., 63
- Hopfield, J., 173
- Hume, D., 191
- Humphreys, P., 191
- Hutter, M., 391,
399, 404, 405
- imprévisibilité, 63
- indécidabilité, 3
- information-
processing
machine, 459
- irréductibilité, 63
- irreversible cellular
automata,
412
- Israeli, N., 65
- Jeu de la Vie, 162,
177
- Kahneman, D., 268
- Kauffman, S., 175,
396
- Kelly, Kevin, 159
- Kleene, S., 19, 513
- Kochen, S., 147
- Kolmogorov, A., 6,
7, 10–13, 19,
22, 25,

- 27–30, 38,
46, 53, 59,
214–216, 249
- Kreisel, G., 21
- Landauer, R., 140,
149, 151,
152, 215, 216
- Langton, C., 225
- Laplace, P.-S., 63
- Leggett, T., xxviii
- Leibniz, G., 3, 89,
191, 317
- Levin, L., 7, 19, 20,
24, 31, 37,
41, 42, 70,
90, 124, 131,
132, 134,
140, 158,
164, 168,
169, 172,
198, 211,
213, 215,
216, 394, 396
- Li, M., 214
- Lloyd, S., iii, v,
xxviii, 50,
91, 109, 111,
125, 170
- Lorentz contraction
hypothesis,
466
- Lorentz transforma-
tions,
493
- Lorenz, E., 63
- Lovelace, A., 433
- Margenstern, M.,
391–393
- Martin-Löf, P., 6,
10, 14, 18,
19, 402, 416,
426
- Maxwell equations,
466
- McAllister, J., 111,
165
- McAllister, J.R.,
xxvii
- McCarthy, J., 328
- McNamara J., 184
- Meyerstein, F., 109,
111, 133,
167, 393
- Minsky, M., 397
- Monte-Carlo
method, 506
- Morange, M., 162
- Newton, I., 63, 89
- noncontinuous
processes,
460
- Pearl, J., 260
- Penrose, R., iii, v,
xxvii, 109,
111, 245
- Planck, M., 141
- plankalkül, 517
- Podolsky, B., 99
- Post tag systems,
513
- Post, E., 19, 417,
418, 513
- punched card, 514
- Putnam, H., 248,
252
- quantum computa-
tion,
411
- Rasoir d’Occam, 9
- Recursive functions,
513
- Reznikova, Z., 195,
266
- Roblot, T., 409
- Rojas, R., 513
- Ronald, E., 191
- Rosen, N., 99
- Rozenberg, G., 392
- Russell, B., 30
- Salomaa, A., 109,
111, 131,
167, 393, 409
- Schönfinkel, M., 513
- Schmidhuber, J., 91,
109, 520
- Schnorr, C., 7, 19
- Schrödinger, E., 400
- Searle, J., 111, 245,
248, 252,
259, 260,
264, 266
- Shannon, C., 7, 8,
12, 139, 140,
145, 146,
151, 186,
201,
203–205,
207,
214–216, 328
- Sipper, M., 191
- SKI combinators,
513
- Solomonoff, R.,
xxvii, 6, 7,
20, 25, 131,
134, 156,
169, 211,
215, 394

- Solovay, R.M., 47
 Specker, E., 147
 Stay, M., 191, 210
 Sutner, K., 69, 420, 426
 Svozil, K., xxviii, 293, 328, 391, 407, 408
 Szudzik, M., 108, 391, 428
 t'Hooft, G., 144, 412
 Tegmark, M., 170
 test de Turing, 219, 259
 Teuscher, C., 391, 399, 400, 408, 409, 413, 414, 428
 Theory of
 Automatons, 457
 Thompson, D'Arcy, 170
 Toffoli, T., xxviii, 152, 153, 216, 519
 Turing, A., iv, vi, 6, 7, 11, 19, 21, 24, 25, 44, 48, 51, 58, 64, 65, 139, 154, 167, 175, 190, 197, 205, 213, 245, 248, 249, 260–264, 267, 316, 391–393, 395, 398, 399, 401, 403, 408, 415, 417, 418, 422, 424, 425, 513–515, 517
 Twardy, C., 165
 Vitányi, P., 214
 von Mises, R., 17
 von Neumann, J., 65, 104, 147, 152, 175, 177, 328, 413, 417, 420, 429, 443, 516
 Weinberg, S., 140
 Wheeler, J., iii, v, 89, 90, 104, 105, 110, 139, 140, 147–150, 215, 401, 430, 435
 Wigner, E., 401
 Wiles, A., 420, 446
 Wolfram, S., iii, v, xxvi, xxviii, 52–54, 57, 64, 66, 90–92, 116, 120, 121, 132, 137–140, 144, 148, 150, 161, 162, 170, 176, 178, 180, 187, 189, 193, 195, 212, 216, 318, 322, 324, 330, 331, 391, 394, 396, 398, 410, 412, 413, 415, 422, 428, 429, 519
 Zeilinger, A., 104, 143, 147
 Zenil, H., 391, 394, 401, 403, 409–416, 418–420, 426, 453, 516
 Zuse, H., 516
 Zuse, K., iii, v, 90, 91, 513, 515, 518
 Calculating
 Space, 453
 Rechnender
 Raum, 453