Tableau Noir

101. — Rached Mneimné. *Réduction des endomorphismes*
102. — Marc Hindry. *Arithmétique*
103. — Jean-Denis Eiden. *Géométrie analytique classique*
104. — Denis Choimet et Hervé Queffélec. *Analyse mathématique.– Grands théorèmes du vingtième siècle*
105. — Pascal Boyer. *Algèbre et géométries*
106. — Patrick Dehornoy. *La théorie des ensembles*
107. — Benoît Monin et Ludovic Patey. *Calculabilité*
108. — Gilles Godefroy. *Introduction aux méthodes de Baire* (à paraître)

Benoît Monin et Ludovic Patey

Calculabilité

Aléatoire, mathématiques à rebours et hypercalculabilité

Calvage & Mounet

BENOÎT MONIN est maître de conférences à l'université de Créteil. Ses travaux portent sur la calculabilité classique, l'aléatoire algorithmique et l'hypercalculabilité.

benoit.monin@computability.fr
https://www.lacl.fr/%7Ebenoit.monin/

LUDOVIC PATEY est chargé de recherche à l'université de Paris. Ses travaux portent sur la calculabilité classique et les mathématiques à rebours.

ludovic.patey@computability.fr
https://ludovicpatey.com/

Mathematics Subject Classification (2010) – Primary :

03-Dxx Computability and recursion theory
03D30 Other degrees and reducibilities in computability and recursion theory
03D60 Computability and recursion theory on ordinals, admissible sets, etc.
03D80 Applications of computability and recursion theory
03D10 Turing machines and related notions
03D20 Recursive functions and relations, subrecursive hierarchies
03D25 Recursively (computably) enumerable sets and degrees
03D32 Algorithmic randomness and dimension
03D55 Hierarchies of computability and definability
03D78 Computation over the reals, computable analysis
03B30 Foundations of classical theories (including reverse mathematics)
68-Qxx Theory of computing
68Q05 Models of computation (Turing machines, etc.)
68Q30 Algorithmic information theory (Kolmogorov complexity, etc.)
03B10 Classical first-order logic
03C07 Basic properties of first-order languages and structures
03F35 Second- and higher-order arithmetic and fragments
03F40 Gödel numberings and issues of incompleteness
03F60 Constructive and recursive analysis

ISBN 978-2-91-635296-1

9 782916 352961

∞ Imprimé sur papier permanent

à nos familles

Table des matières

4. Degrés Turing

5. Hiérarchie arithmétique

6. La thèse de Church-Turing

7. Immunité et croissance de fonction

8. Classes Π^0_1 et degrés PA

Préambule

Ce livre est une introduction à la théorie de la calculabilité ainsi qu'à trois de ses ramifications principales, à savoir la théorie algorithmique de l'aléatoire, les mathématiques à rebours et l'hypercalculabilité. Il s'agit d'un ouvrage principalement destiné aux étudiants et enseignants en master de recherche en informatique et mathématiques, ainsi qu'aux chercheurs désirant acquérir une solide connaissance en calculabilité.

Raison d'être du livre

Cet ouvrage trouve ses origines dans plusieurs constats.

- ▷ La littérature francophone sur la calculabilité classique est quasiment inexistante. Il existe quelques introductions à la calculabilité en français qui s'intègrent dans le cadre d'une présentation de la théorie de la complexité, comme l'excellent ouvrage d'Olivier Carton *Langages formels, calculabilité, et complexité* [28]. Les modèles de calcul sont alors présentés comme une base solide pour développer une théorie de la complexité algorithmique. Les résultats de calculabilité à proprement parler dépassent rarement l'indécidabilité du problème de l'arrêt et la définition de la hiérarchie arithmétique.
- ▷ Il existe de nombreux livres de référence en anglais sur la calculabilité (*Classical Recursion Theory : The Theory of Functions and Sets of Natural Numbers* de Piergiorgio Odifreddi [169], *Computability Theory* de Barry Cooper [41] ou encore *Turing computability : Theory and Applications* de Robert Soare [208]). Concernant la théorie algorithmique de l'aléatoire, on citera *Computability and Randomness* d'André Nies [167], et *Algorithmic Randomness and Complexity* de Rodney

Downey et Denis Hirschfeldt [50]. En mathématiques à rebours, la référence historique est *Subsystems of Second Order Arithmetic* de Stephen Simpson [203]. On mentionnera l'ouvrage plus récent de Denis Hirschfeldt, *Slicing the Truth* [87], et l'ouvrage en cours de rédaction, *Reverse Mathematics* de Damir Dzhafarov et Carl Mummert. Enfin, en hypercalculabilité, les deux références sont *Higher recursion theory* de Gerald Sacks [192] et *Recursion Theory : Computational Aspects of Definability* de Chi Tat Chong et Liang Yu [35]. Chacun de ces ouvrages présente l'état de l'art de la recherche pour un sous-domaine spécifique de la calculabilité, mais il n'existe pas un unique livre qui organise une présentation cohérente de ces différents aspects.

▷ Le calculabilité est un sujet d'étude extrêmement vaste, comme le laisse entrevoir la profusion d'articles sur le sujet et la taille des ouvrages de référence en anglais. Cependant, ce domaine reste très peu représenté en France, aussi bien du point de vue de la recherche que de l'enseignement. Peu de cours de calculabilité sont proposés en master, et leur contenu tend à véhiculer l'idée fausse que la calculabilité est une théorie des modèles de calcul. Cet ouvrage a pour ambition de donner ses lettres de noblesse à la calculabilité en France en donnant un petit panorama de sa grande richesse, encore trop méconnue.

Organisation du livre

Ce livre est structuré en quatre grandes parties, à savoir la calculabilité classique, l'aléatoire algorithmique, les mathématiques à rebours et l'hypercalculabilité.

Plan général

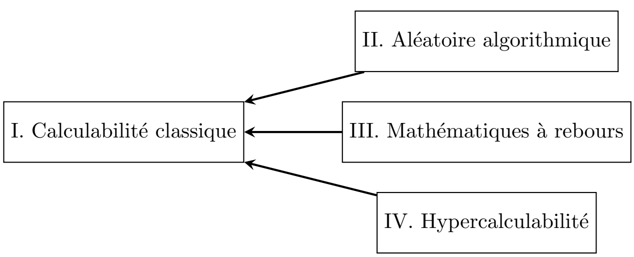
▷ La *calculabilité classique* est l'étude des degrés Turing, autrement dit la puissance calculatoire des ensembles d'entiers naturels. Elle constitue le cœur historique de la calculabilité, et le socle épistémique sur lequel s'appuient les trois parties suivantes.

▷ L'*aléatoire algorithmique* utilise la calculabilité classique pour donner un cadre effectif à la théorie de la mesure, qui permet d'étudier individuellement les suites de bits dites « aléatoires ». Les hiérarchies induites par la calculabilité classique permettent de définir différents niveaux d'aléa, à la lumière desquels on peut, par exemple, réexaminer la signification de tel ou tel théorème probabiliste stipulant qu'une propriété est vraie presque partout.

▷ Les *mathématiques à rebours* forment un programme sur les fondements des mathématiques, qui vise à identifier les axiomes nécessaires pour prouver les théorèmes mathématiques de la vie de tous les jours [1]. Elles reposent sur une théorie de base, RCA_0, dont les axiomes capturent les mathématiques « calculables » grâce à une correspondance entre la calculabilité et la définissabilité par des formules logiques.

- ▷ L'*hypercalculabilité* étend la notion de calculabilité à un cadre plus général rejoignant la théorie des ensembles. De la même manière que les opérations arithmétiques élémentaires s'étendent aux ordinaux, les machines de Turing peuvent étendre leur temps de calcul, des entiers aux ordinaux, et manipuler ainsi de plus grandes classes de réels. Il s'agit de l'approche par « modèle de calcul » de l'hypercalculabilité, qui correspond, comme pour la calculabilité classique, à certaines classes logiques.

Les trois dernières parties s'appuient toutes fortement sur la calculabilité classique, mais sont relativement indépendantes entre elles, et peuvent être pour l'essentiel, lues dans un ordre quelconque :



Dépendances entre les quatre grandes parties du livre

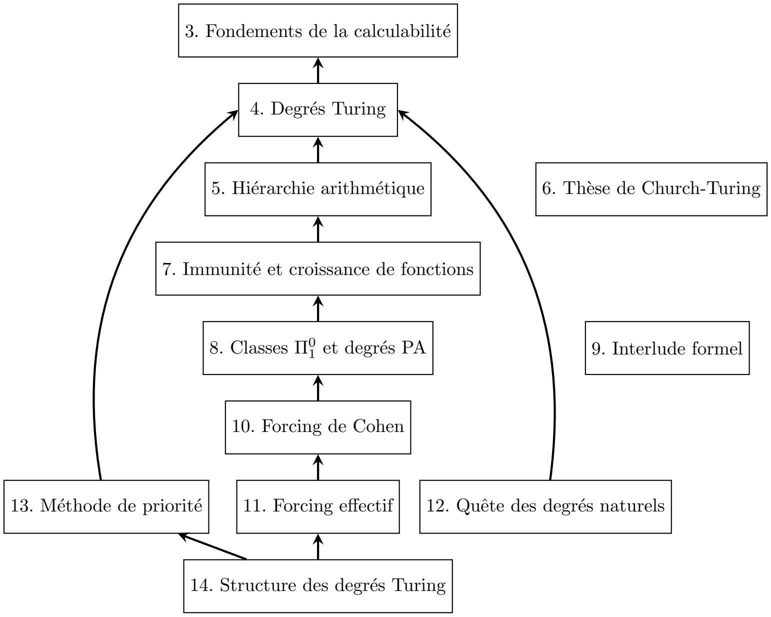
Notons tout de même la notion de « classe Borélienne » introduite dans la partie II, et fondamentale pour la compréhension de la partie IV.

Calculabilité classique

La calculabilité classique a un rôle prépondérant en ce qu'elle fixe un cadre formel et une série d'outils qui serviront à développer les parties suivantes. Il convient donc de s'attarder sur la première partie et de détailler les dépendances de ses chapitres. Les chapitres fondamentaux sont principalement à lire linéairement, avec les exceptions suivantes : les chapitres 6 et 9 peuvent être lus indépendamment des autres, mais seront cependant utiles pour aborder sereinement la partie III du livre, sur les mathématiques à rebours. Les chapitres 12 et 14 ne seront pas absolument nécessaires pour la compréhension des parties suivantes, et visent à prendre un peu de recul

1. De la vie de tous les jours du mathématicien, s'entend.

sur notre travail. Le chapitre 12 — moins technique — le fera à travers l'examen d'une question spécifique, à la frontière de la philosophie ; et le chapitre 14 à travers une étude plus abstraite de la structure des degrés Turing, et la présentation de quelques-unes des grandes questions ouvertes du domaine.



Dépendances entre les chapitres de la calculabilité classique

Comment lire ce livre

Pour les enseignants

Ce livre peut servir de support pour un cours d'introduction à la calculabilité de niveau master, ainsi que pour des cours thématiques plus avancés, parlant de théorie algorithmique de l'aléatoire, de mathématiques à rebours et d'hypercalculabilité. Les sujets abordés vont bien au-delà des connaissances en calculabilité que l'on peut attendre d'un étudiant en master. Nous allons donc proposer un plan de cours contenant les notions incontournables.

L'équivalence entre les modèles de calcul forme un socle robuste pour le développement de la calculabilité. Cependant, les preuves peuvent sembler assez longues et fastidieuses. De nos jours, avec la démocratisation des ordinateurs, on peut s'attendre à ce que les étudiants possèdent une certaine intuition de ce qu'est un algorithme, et il semble préférable de partir de cette intuition pour faire les premiers développements afin d'éviter un formalisme relativement lourd. Nous recommandons d'aborder l'équivalence des modèles de calcul, notamment entre les fonctions générales récursives et les machines de Turing, à travers une séance de travaux dirigés, où les étudiants auront l'occasion de manipuler les formalismes en définissant des fonctions de plus en plus compliquées pour se convaincre que ces définitions permettent de capturer tous les algorithmes.

Nous invitons nos lecteurs à suivre les développements des différents chapitres 3, 4, 5, 7, 8, 10, 13 dans cet ordre. Le chapitre 3 établit les premiers théorèmes fondamentaux de la calculabilité sur la base de l'intuition que l'on a des algorithmes. Le chapitre 4 définit la notion de machine à oracle, et de degré Turing. On y trouve les définitions les plus centrales de la calculabilité, comme le saut de Turing, et la méthode des extensions finies qui est une technique très puissante pour prouver l'existence de certains degrés Turing. Le chapitre 5 sur la hiérarchie arithmétique établit un lien essentiel entre la puissance calculatoire d'ensembles d'entiers et leur définissabilité par des formules de l'arithmétique, à travers le théorème de Post.

Avec le chapitre 7, on commence l'étude de différentes propriétés calculatoires fondamentales, comme la notion de degré hyperimmune, et ses liens avec l'existence de fonctions à croissance rapide. Cette étude est poursuivie dans le chapitre 8 sur les classes Π^0_1, où l'on définit la notion de degré PA qui est une notion centrale en calculabilité. On la retrouve tout au long de ce livre, notamment en aléatoire algorithmique et en mathématiques à rebours.

Le chapitre 10 introduit une technique fondamentale de la calculabilité, à savoir le forcing, présentée ici comme une élaboration de la méthode des extensions finies du chapitre 4. Ce chapitre peut également servir comme le premier niveau d'une compréhension graduelle de la technique générale du forcing de la théorie des ensembles.

Le chapitre 13 introduit finalement les méthodes de priorités, une autre technique fondamentale de la calculabilité, qui permet notamment de montrer l'existence de certains degrés calculatoirement énumérables.

Pour les étudiants

Les compétences requises pour comprendre les notions présentées dans cet ouvrage sont celles d'une première année de licence en informatique ou mathématiques. Il est nécessaire pour l'essentiel de comprendre le langage mathématique usuel (variables, quantifications, etc.), d'avoir quelques notions élémentaires de logique (preuve par contraposée, preuve par l'absurde, etc.), et de comprendre les éléments du corpus mathématique de base (comprendre ce qu'est une bijection, ce qu'est une intersection entre deux ensembles, les fonctions puissance et logarithme, etc.). En plus de cela, une expérience même sommaire en programmation, ou à défaut la compréhension de ce qu'est un algorithme, est également nécessaire pour aborder sereinement la lecture de ce livre.

Les mathématiques que nous utiliserons et qui ne sont pas enseignées dans une première année de licence seront introduites et expliquées au fur et à mesure des besoins (notions de base de topologie ou de théorie de la mesure par exemple). Ayant établi cela, notons tout de même que le degré d'élaboration des preuves, ainsi que la technicité de certains concepts, rendront sans doute cet ouvrage difficile à aborder sans une certaine maturité mathématique.

Les techniques développées en calculabilité sont assez différentes de celles apprises à travers un cursus mathématique standard. Cette particularité de la calculabilité est une force et rend cette discipline plus accessible puisqu'elle est peu sensible aux lacunes que l'on peut avoir développé au cours de son cursus (ou de son absence de cursus). En revanche, cette différence peut également déstabiliser l'étudiant car cela demande de se créer un univers conceptuel. Il va sans dire qu'en l'absence de professeur, il est d'autant plus essentiel de faire les exercices proposés au cours du livre pour bien intégrer les concepts. Les solutions sont disponibles à la fin du livre.

La taille de cet ouvrage peut s'avérer décourageante pour un étudiant désireux de faire ses premiers pas en calculabilité. Nous rappelons que ce livre couvre des connaissances allant bien au-delà de ce que l'on attend d'un étudiant en master. Nous recommandons donc aux autodidactes de suivre la suggestion de cours de la section précédente, destinée aux enseignants.

Pour les chercheurs

Cet ouvrage est une introduction à la calculabilité et à plusieurs de ses branches principales. Il ne faut cependant pas s'arrêter à l'aspect introductif de cet ouvrage, car la plupart des résultats présentés correspondent à l'état de l'art de la recherche. Ce livre s'adresse donc aussi bien aux chercheurs de domaines connexes désireux d'acquérir de solides connaissances en calculabilité, qu'aux doctorants et chercheurs se destinant à la recherche en calculabilité. En effet, les techniques et concepts de ce livre rendent directement accessibles les articles de recherche du domaine.

Exercices

Les chapitres sont parsemés d'exercices de difficulté variable, dont la correction est donnée à la fin du livre. On ne rappellera jamais assez l'importance de faire des exercices pour bien assimiler les notions présentées dans les chapitres. L'intuition des concepts se crée en les manipulant sous toutes leurs formes. La difficulté des exercices est indiquée à l'aide d'un système d'étoiles ($\star$) allant de 0 à 3 : un exercice sans étoile est une application directe des définitions, tandis qu'un exercice à deux étoiles demande une profonde maîtrise des concepts pour être résolu. On trouvera également quelques exercices à trois étoiles, qui sont d'un niveau « recherche ».

À travers cet ouvrage, nous allons présenter beaucoup de propriétés calculatoires sur les ensembles d'entiers ou sur d'autres structures plus complexes. Outre les exercices donnés, il est important de faire preuve d'une curiosité intellectuelle consistant à chercher systématiquement comment se combinent ces propriétés, savoir si l'on peut construire des objets satisfaisant plusieurs d'entre elles simultanément, et ainsi de suite. De même, lorsque les théorèmes possèdent des hypothèses, il est utile de chercher des contre-exemples sans ces hypothèses, afin de mieux comprendre leur nécessité ainsi que leur usage dans la preuve.

Errata

Il est impossible d'écrire un livre de cette taille sans laisser se glisser un certain nombre de coquilles. Cet ouvrage ne dérogera probablement pas à cette règle. Nous maintiendrons une liste des coquilles sur la page web des auteurs. Vous pouvez signaler les erreurs à l'une des adresses suivantes :

`benoit.monin@computability.fr`
ou
`ludovic.patey@computability.fr`

Remerciements

Notre tout premier remerciement va à l'Agence nationale de la recherche, qui a en grande partie financé la collaboration entre les deux auteurs, notamment pendant l'écriture du livre, dans le cadre du projet « Aspects Calculatoires des Théorèmes Combinatoires – ACTC »

`https://anr.fr/Projet-ANR-19-CE48-0012`

Plusieurs résultats présentés dans ce livre, notamment le théorème 25-3.23 sur le principe des tiroirs ou la preuve simplifiée du théorème 25-3.24 de Liu proviennent d'articles financés par ce même projet.

Nous tenons également à remercier nos équipes et organismes de rattachement, qui nous ont fourni un vivier d'émulation intellectuelle ainsi qu'un soutien moral et financier. Pendant la rédaction de cet ouvrage, Monin était maître de conférences dans le Laboratoire d'Algorithmique, Complexité et Logique de l'Université Paris-Est Créteil et Patey chargé de recherche au CNRS, dans l'équipe Algèbre, Géométrie, Logique de l'Institut Camille Jordan à Villeurbanne.

De nombreux collègues nous ont généreusement aidé à améliorer la qualité de cet ouvrage, en donnant un regard critique sur son contenu scientifique et pédagogique, fort de leurs expertises respectives, ou bien en signalant les erreurs typographiques qui se sont inévitablement glissées dans le livre. Nous remercions donc Paul-Elliot Anglès d'Auriac, Sébastien Tavenas, Pascal Vanier, Mathieu Hoyrup, Benjamin Hellouin, Laurent Bienvenu, Denis Kuperberg, Julien Cervelle, Damir Dzhafarov, Loïc Gassmann, William Gaudelier, Denis Hirschfeldt, Quentin Le Houerou, Alexander Shen, Keita Yokoyama, Adrien Deloro, Pascal Monin et Shahin Amini.

Le contenu scientifique du livre est avant tout l'œuvre de la communauté de la calculabilité. Les auteurs ont forgé leurs intuitions en lisant les ouvrages de leurs prédécesseurs et en ajoutant leur pierre à ce magnifique édifice intellectuel. Nous tenons à remercier nos collègues au niveau international pour les collaborations et les visites mutuelles ayant permis d'améliorer notre compréhension du sujet.

Nous tenons également à remercier Laurent Bienvenu, qui par son travail et à travers la direction de nos thèses respectives, a su nous transmettre sa passion, et a largement contribué à introduire la calculabilité en France.

La rédaction d'un ouvrage de cette ampleur prend beaucoup de temps et d'énergie, et n'aurait pu avoir lieu sans le soutien moral de nos familles et amis.

Nous tenons enfin à remercier les éditions Calvage et Mounet pour leur confiance, leur soutien et leur travail éditorial, et particulièrement Rached Mneimné pour sa relecture minutieuse du livre.

Chapitre 1

Introduction

> Le savant n'étudie pas la nature parce que cela est utile ; il l'étudie parce qu'il y prend plaisir et il y prend plaisir parce qu'elle est belle. Si la nature n'était pas belle, elle ne vaudrait pas la peine d'être connue, la vie ne vaudrait pas la peine d'être vécue. Je ne parle pas ici, bien entendu, de cette beauté qui frappe les sens, de la beauté des qualités et des apparences ; non que j'en fasse fi, loin de là, mais elle n'a rien à faire avec la science ; je veux parler de cette beauté plus intime qui vient de l'ordre harmonieux des parties, et qu'une intelligence pure peut saisir.
>
> Science et méthode, Henri Poincaré

Qu'est-ce que la calculabilité ? On considère classiquement la calculabilité comme l'un des quatre piliers de la logique, aux côtés de la théorie des ensembles, la théorie des modèles et la théorie de la preuve. Le domaine s'est au départ forgé sur la question de ce qui caractérise les fonctions $f : \mathbb{N} \to \mathbb{N}$ dont les valeurs peuvent être obtenues par un processus purement *mécanisable* ou *algorithmique*, en un temps fini, bien qu'arbitrairement grand. Nous dirons que de telles fonctions sont *effectivement calculables*. Bien avant l'apparition des premiers ordinateurs, la calculabilité a fondé sa base théorique sur un constat — ou plutôt un miracle — à savoir l'existence d'une définition robuste, consensuelle et indépendante de tout formalisme, de la notion épistémologique de fonction effectivement calculable.

La question initiale — à savoir « Qu'est-ce qu'une fonction calculable ? » — ayant obtenu une réponse satisfaisante, l'étude s'est naturellement portée vers la question de savoir, parmi les fonctions naturelles, lesquelles sont calculables et lesquelles ne le sont pas. Par la suite, le domaine a connu un développement considérable grâce à la notion de calculabilité relative, la question n'étant plus de déterminer si une fonction est calculable ou non, mais d'identifier la puissance calculatoire intrinsèque à cette fonction, à travers des questions comme « Si cette fonction était calculable, quelles autres fonctions pourrait-on calculer ? »

Plus récemment, le sujet d'étude s'est étendu à de très nombreux objets mathématiques — par exemple des structures algébriques ou des sous-ensembles de $\mathbb{R}$ — et a donné de nombreuses ramifications. Nous verrons notamment dans ce livre que la calculabilité sert de fondement robuste à la théorie algorithmique de l'aléatoire, et aux mathématiques à rebours, dont les objets d'études sont les théorèmes mathématiques eux-mêmes.

De nos jours, l'appellation « calculabilité » pour un domaine qui étudie des objets mathématiques arbitraires, dont la plupart ne sont pas calculables, peut sembler étonnante, voire un reliquat de son sujet d'étude historique. En vérité, ce nom est toujours judicieux, mais sa signification a changé : le terme *calculabilité* ne porte plus sur le sujet de l'étude, mais sur l'angle sous lequel le sujet est abordé. Une définition moderne de la calculabilité en une phrase pourrait être : **la calculabilité est l'étude des mathématiques sous le prisme de leur complexité calculatoire**.

1. Qu'est-ce qu'une fonction calculable ?

La principale difficulté de cette question réside dans l'obtention d'une classe de fonctions suffisamment robuste pour ne pas dépendre du modèle d'ordinateur, du choix du langage de programmation, des progrès technologiques, ou de l'avancée de la connaissance de manière générale.

Avec l'avènement des ordinateurs, la notion d'algorithme s'est peu à peu ancrée dans la culture scientifique. Toute personne ayant déjà eu un premier contact avec la programmation se sera déjà formée une bonne idée de ce qu'est une tâche automatisable. Forts de notre connaissance de l'informatique, la définition suivante viendrait naturellement : « Une fonction est effectivement calculable si elle possède un algorithme, autrement dit si elle peut être programmée dans un langage suffisamment expressif, et exécutée par un ordinateur suffisamment puissant. »

Cette définition, si elle a l'avantage d'être en adéquation avec notre intuition, ne fournit pas un cadre suffisamment formel pour raisonner sur la classe des fonctions calculables. Une seconde approche consisterait à fixer un ordinateur et un langage de programmation étalon, et définir une fonction comme calculable si elle est programmable dans ce langage, et exécutable par cet ordinateur en temps fini, à l'aide de suffisamment de mémoire. Si l'on ne se préoccupe pas de la rapidité d'exécution, ni de l'espace mémoire nécessaire, il apparaît rapidement que cette définition coïncide avec la précédente. En effet, la puissance et la mémoire des ordinateurs augmentent au gré des progrès technologiques, et permettent donc d'exécuter plus rapidement les programmes, mais n'augmentent pas pour autant la classe des fonctions calculables. Même les ordinateurs basés sur de nouveaux paradigmes de calcul, comme les ordinateurs quantiques ou biologiques, sont simulables — au prix d'un surcoût d'espace et de temps exponentiel — par des ordinateurs classiques, et ne changent donc pas la classe des fonctions calculables. Quant aux langages de programmation, l'existence de systèmes d'exploitation et d'interpréteurs permettent de se convaincre aisément que les principaux d'entre eux tels que C++, Java ou Python, permettent de programmer — plus ou moins élégamment — les mêmes fonctions mathématiques. Cela montre donc empiriquement une certaine robustesse dans la définition de la classe des fonctions programmables.

Un problème subsiste : quelle est la garantie que les ordinateurs actuels représentent la limite de ce qui est automatisable, ou calculable par un être humain ? Qui nous dit qu'avec les progrès de la science, nous ne découvrirons pas un nouveau paradigme de calcul ou une nouvelle manière de raisonner permettant de considérer comme calculable une plus large classe de fonctions ? C'est là le sujet d'une longue quête fondationnelle débutée au XXe siècle, et aboutissant à la fameuse thèse de Church-Turing en 1936, que nous présenterons dans le chapitre 6.

2. Quelles sont les fonctions incalculables ?

Au regard de notre définition précédente, pour l'instant très informelle, la plupart — si ce n'est la totalité — des fonctions mathématiques utilisées au quotidien sont calculables : l'addition, la multiplication, la fonction $(n, m) \mapsto n^m$, la fonction qui à n associe le n-ième nombre premier, ou encore celle qui calcule le plus grand diviseur commun de deux entiers naturels, sont toutes calculables. On peut rajouter à cette liste des exemples moins triviaux : la fonction qui prend un programme informatique écrit en C++ et détermine si le programme est syntaxiquement correct — c'est ce que fait entre autres choses un compilateur pour C++ — ou encore celle qui renvoie la n-ième décimale de π, $\sqrt{2}$ ou du nombre d'or — chacun de ces nombres est la somme d'une suite calculable de rationnels de convergence suffisamment rapide — ou pour finir celle qui à n associe le nombre de parties possibles que l'on peut faire au jeu de go sur un plateau — appelé aussi *goban* — de taille $n \times n$. Ce dernier exemple illustre en particulier le fait suivant : on ne s'occupe pas en calculabilité du temps que prend un calcul. Seule l'existence d'un algorithme nous importe. Dans le cas du nombre de parties au jeu de go, l'algorithme en question repose sur une idée simple ; il « suffit » de lister toutes les parties possibles et de les compter. Cependant, le temps d'exécution d'un tel algorithme est tellement grand que cela le rend impossible à utiliser en pratique pour $n > 2$ ([1]). Pour $n = 19$, qui est la taille d'un goban standard, ce nombre est compris entre $10^{10^{48}}$ et $10^{10^{171}}$ [225], ce qui fait clairement trop de parties à compter même si tous les ordinateurs de monde s'y attelaient pour un milliard d'années...

Même si la fonction de multiplication par 2 nous est, en un sens, bien plus accessible que celle qui compte le nombre de parties au jeu de go, il existe un algorithme qui calcule chacune d'entre elles. Ces deux fonctions ne sont donc pas différentes l'une de l'autre du point de vue de la calculabilité : elles sont toutes les deux calculables, et nous allons principalement nous intéresser aux fonctions qui *ne le sont pas*, c'est-à-dire les fonctions dont les valeurs *ne peuvent pas* être obtenues par un processus purement mécanisable ou algorithmique. La simple existence de telles fonctions n'est pas une évidence en soi, et l'une des premières tâches à laquelle nous nous attellerons sera d'en montrer l'existence. Cela sera fait dans le chapitre suivant via l'argument diagonal de Cantor. Nous donnerons ensuite tout au long du livre de très nombreux exemples de telles fonctions, la plus connue d'entre elles étant sans doute le *problème de l'arrêt*, défini comme la fonction qui prend en entrée un programme, et détermine si son exécution va s'arrêter, en temps fini nécessairement. Nous verrons que le problème de l'arrêt n'est pas calculable ; et il est important de comprendre qu'il s'agit bien ici d'une impossibilité théorique et fondamentale, qui ne dépend pas de la puissance ou vitesse de calcul des ordinateurs. L'incalculabilité du problème de l'arrêt n'est pas due à une ignorance de son algorithme qui pourrait être un jour découvert, mais bien à une impossibilité absolue, car l'existence d'un tel algorithme entraînerait un paradoxe.

3. Motivations

La calculabilité porte principalement sur l'étude des fonctions — ou objets mathématiques plus généraux — incalculables. Il est légitime de se demander si une telle étude est bien raisonnable. Si même certaines fonctions calculables nous sont inaccessibles — comme le nombre de parties possibles

1. Il y a déjà 386 356 909 593 parties possibles sur un goban de taille 2×2 [225] !

au jeu de go — alors à quoi bon se donner la peine de réfléchir sur des fonctions *encore plus inaccessibles* ?

Une première motivation pour notre étude est d'ordre exploratoire. Il existe des objets inaccessibles, essayons d'en explorer l'univers. Simplement parce qu'il est là, et par curiosité sur les mystères qu'il renferme. Nos efforts seront récompensés par une série de théorèmes d'une très grande profondeur. Quiconque se plonge avec sérieux dans les développements de ce livre, une fois peut-être passé quelques difficultés d'adaptation inhérentes à toute discipline scientifique, verra un monde d'une richesse stupéfiante prendre vie dans son esprit, avec sa faune et sa flore, ses règles et mécanismes. La calculabilité est caractérisée par la nature très dynamique de ses preuves, chacune d'elles en offrant un aperçu sur le fonctionnement détaillé d'un fragment de la machinerie titanesque qui anime cet univers.

Une deuxième motivation survient tout simplement par nécessité. Les Pythagoriciens se sont retrouvés contraints et forcés d'admettre l'existence de mesures irrationnelles, comme la diagonale d'un carré de côté 1, ce qui allait à l'encontre de leur compréhension du monde, qu'ils pensaient explicable uniquement en se fondant sur les rapports entre nombres entiers. Mais si l'on admet l'existence des entiers, et l'existence du carré, on est forcé d'admettre celle de quantités *incommensurables*, que l'on appelle aujourd'hui irrationnelles, comme $\sqrt{2}$. De la même manière, nous verrons que si l'on admet l'existence des objets calculables, on est forcé aussi d'admettre l'existence d'objets qui ne le sont pas, et qui apparaissent malgré tout naturellement dans toute une série de situations.

Une dernière motivation enfin est d'ordre pratique. La calculabilité, via la compréhension qu'elle donne des objets incalculables, a obtenu des succès majeurs en fournissant un cadre formel pour l'étude de questions à la frontière entre science et philosophie. Nous en verrons deux : la recherche de la définition d'objets aléatoires avec la partie II, et la compréhension de ce que signifie *la force* d'un théorème, notamment par rapport à un autre, avec la partie III. La partie IV de ce livre amènera quant à elle la calculabilité vers la frontière qu'elle partage avec la théorie des ensembles.

4. Panorama de la calculabilité

La calculabilité peut se décomposer en plusieurs sous-domaines, qui s'appuient tous sur la même notion robuste de fonction effectivement calculable.

4.1. Domaines couverts par ce livre

Cet ouvrage est décomposé en quatre parties, chacune d'entre elles couvrant une ramification de la calculabilité : la calculabilité classique, l'aléatoire algorithmique, les mathématiques à rebours, et l'hypercalculabilité.

Calculabilité classique

Comme nous l'avons spécifié, la calculabilité porte avant tout sur des objets que l'on ne peut pas calculer. La calculabilité classique se concentre sur les fonctions $f : \mathbb{N} \to \mathbb{N}$ ainsi que sur les ensembles d'entiers $E \subseteq \mathbb{N}$. Remarquons qu'un tel ensemble peut aussi être représenté par sa fonction caractéristique $\chi_E : \mathbb{N} \to \mathbb{N}$ définie par $\chi_E(x) = 1$ si $x \in E$, et $\chi_E(x) = 0$ sinon.

Les développements de la calculabilité classique s'articulent autour d'un outil fondamental qui nous permettra de comparer ou encore de mesurer le *degré d'incalculabilité* d'une fonction, appelé aussi *degré d'insolubilité* ou encore *degré Turing*, en référence au mathématicien Alan Turing qui introduisit la notion. Fixons une fonction non calculable $g : \mathbb{N} \to \mathbb{N}$. Il est naturel de se demander « Si j'étais capable de calculer g, quelles autres fonctions pourrais-je calculer ? » On dit qu'une fonction $f : \mathbb{N} \to \mathbb{N}$ est *calculable relativement à g* (ou g-calculable) s'il existe un algorithme permettant de calculer f dans un langage de programmation étendu, où l'on aurait rajouté la fonction g comme primitive : une instruction spéciale nous permet d'appeler la fonction g sur un paramètre n dans notre programme, comme si elle existait réellement, et d'en récupérer le résultat. Si f est g-calculable, rien ne nous indique comment calculer g, mais si l'on disposait d'un « oracle » nous permettant de calculer les valeurs de g, il serait possible de calculer les valeurs de f.

Cette notion de calculabilité relative nous permet de définir un pré-ordre partiel entre les fonctions, notant $f \leqslant_T g$ si la fonction f est g-calculable. Il s'agit de la *réduction Turing*. Différentes fonctions peuvent porter la même puissance calculatoire, au sens où elles sont mutuellement calculables. On définit donc le *degré Turing* d'une fonction $f : \mathbb{N} \to \mathbb{N}$ l'ensemble $\deg_T f$ de toutes les fonctions g telles que $f \leqslant_T g$ et $g \leqslant_T f$. La notion de degré Turing représente une puissance calculatoire, au sens où deux fonctions de même degré Turing sont indistinguables du point de vue de la calculabilité. Le pré-ordre partiel sur les fonctions induit un ordre partiel sur les degrés Turing.

La calculabilité classique porte principalement sur l'étude des degrés Turing munis de la relation d'ordre partielle définie ci-dessus. Existe-t-il une infinité de puissances calculatoires ? Sont-elles linéairement ordonnées ? Plus généralement, quelles sont les propriétés de cet ordre partiel ? Il s'avère que cette structure est extrêmement riche et complexe, comme nous aurons l'occasion de le voir.

Aléatoire algorithmique

La théorie classique des probabilités étudie les phénomènes probabilistes, modélisés avec succès via la notion de mesure qui sert à définir formellement les lois de probabilité. Cette théorie n'a en revanche pas les outils nécessaires — et ce n'est pas là son objectif — pour parler d'objets aléatoires *individuellement*. C'est ce point précis que la théorie algorithmique de l'aléatoire se propose d'éclaircir, en s'appuyant sur la calculabilité. Voyons à travers un exemple de quoi il s'agit.

Représentons un nombre réel $R \in [0,1]$ par son développement binaire, de la forme $R = 0.b_0b_1b_2b_3\dots$ où $(b_n)_{n\in\mathbb{N}}$ est une suite de bits. Supposons que le réel R soit obtenu en tirant ses bits *au hasard* par une suite de tirage à pile ou face. On suppose bien entendu que chaque tirage est *équiprobable* : nous avons une chance sur deux d'obtenir pile et une chance sur deux d'obtenir face. L'intuition nous dicte que le réel R ainsi obtenu est *aléatoire*. Que cela signifie-t-il exactement ? On ne s'attend par exemple pas à n'obtenir que des « pile » sur les cent mille premiers lancers : si chaque tirage est équiprobable, cela ne peut arriver, ou en tout cas avec une probabilité tellement faible que l'on peut la considérer comme négligeable. On ne s'attend pas non plus à obtenir deux fois plus de « pile » que de « face ». Là encore, la probabilité que cela arrive sur cent mille tirages est tellement faible que l'on supposera les tirages biaisés plutôt que d'être témoin d'un événement si improbable. On peut de fait identifier une première propriété que l'on est en droit d'attendre d'une suite de tirages équiprobables : la suite obtenue devrait respecter la loi des grands nombres, c'est-à-dire que les nombres de tirages « pile » et « face » doivent à peu près être les mêmes.

Cela est-il pour autant suffisant ? Supposons à présent par exemple que sur chaque tirage numéro n, si n est un nombre premier on obtient systématiquement un « pile ». Dans l'hypothèse où une certaine obsession des nombres premiers nous conduirait à remarquer ce curieux phénomène, nous serons là encore en face d'une énigme — un peu absurde — et l'on sera amené à penser que d'une manière ou d'une autre, quelque chose d'anormal est en train de se produire. Mais prenons encore plus de recul. Au fond, et peu importe la suite de bit obtenue, on peut identifier des nombres $n_1 < n_2 < n_3 < \dots$ tels que les tirages numéros n_1, n_2, n_3, $\dots$ sont tous des tirages « pile ». Dans le cas où notre suite n_1, n_2, n_3, $\dots$ contient les nombres premiers, cela nous semble relever d'un « bug probabiliste », mais pourquoi cela devrait-il être acceptable si n_1, n_2, n_3, $\dots$ sont des nombres entiers quelconques ? C'est là que la calculabilité entre en jeu, et va nous permettre de formaliser précisément les propriétés que devrait avoir — en accord avec notre intuition humaine — une suite de bits aléatoire.

Mathématiques à rebours

La notion de *théorème* est relative à un système d'axiomes. Lorsque l'on omet de mentionner le système de référence, il est communément admis que l'on se réfère au système de Zermelo Fraenkel (ZF), qui représente un ensemble d'axiomes consensuels servant de fondement à l'ensemble des mathématiques. Le système ZF est cependant très puissant, et nous n'avons aucune garantie de son absence de contradiction.

Les mathématiques à rebours visent à trouver les axiomes nécessaires et suffisants pour prouver les théorèmes des mathématiques de tous les jours. Il s'agit donc d'étudier des théorèmes existants, pour en trouver des preuves plus élémentaires, ou au contraire pour montrer l'optimalité de leur preuve. Mieux comprendre les hypothèses des théorèmes permet de mieux maîtriser leur « fragilité » face à une potentielle contradiction du système de preuves. Il s'agit donc d'une démarche méta-mathématique visant à répondre à la question « Quelle confiance peut-on avoir en nos mathématiques ? »

Au premier abord, cette démarche n'est pas liée à la calculabilité. Cependant, les mathématiques à rebours se réfèrent à une théorie de base, RCA_0, capturant les *mathématiques calculables*, et qui représente une base de confiance plus en lien avec le monde concret, car ses objets étant calculables, ils peuvent être représentés par un algorithme, donc possèdent une description finitaire. Les mathématiques à rebours consistent donc, étant donné un théorème T, à chercher des axiomes A tels que RCA_0 prouve l'équivalence entre A et T. Par le choix de la théorie de base RCA_0, les équivalences sont des procédés calculatoires faisant appel aux outils de la calculabilité.

Hypercalculabilité

Une des raisons du succès de la calculabilité en tant qu'outil d'analyse des mathématiques réside dans l'existence d'une solide intuition de la notion de calcul, permettant ainsi de guider la manipulation des concepts et de prouver des théorèmes sans s'embarrasser d'un lourd formalisme. L'hypercalculabilité vise à étendre la portée de ces outils à des modèles de calcul plus puissants, qui peuvent être vus comme des machines ayant la possibilité de poursuivre leur exécution pendant un temps de calcul infini (formellement en temps de calcul ordinal). Tout comme les notions de calculabilité classique peuvent être capturées par des formules logiques, il en va de même pour l'hypercalculabilité. Par exemple, là où les ensembles d'entiers que l'on peut énumérer (dans le désordre) par un programme informatique sont ceux qui peuvent être décrits par une formule dite Σ^0_1 de l'arithmétique, ceux qui sont énumérables par un programme informatique hypercalculable sont ceux qui peuvent être définis par une formule dite Π^1_1 de l'arithmétique.

Nous verrons que cet aspect des choses rapproche l'hypercalculabilité de la théorie descriptive des ensembles, une branche de la théorie des ensembles qui classifie ces derniers selon le degré de difficulté à les décrire. L'hypercalculabilité peut être vue comme un pont entre la théorie descriptive des ensembles et la calculabilité classique.

4.2. Autres branches de la calculabilité

Afin de permettre à cet ouvrage de conserver une taille raisonnable, nous avons fait le choix de faire l'impasse sur deux branches importantes de la calculabilité, à savoir la théorie des structures calculables et l'étude des degrés d'énumération.

Théorie des structures calculables

Il s'agit d'une branche de la calculabilité qui étudie dans quelle mesure les propriétés algébriques d'une structure mathématique affectent leur complexité descriptive. Par structure, on entend des ensembles munis d'opérations, comme les groupes, les anneaux et les corps, mais également toute structure au sens de la théorie des modèles. Cette branche emprunte ses techniques à la fois à la théorie des modèles et à la calculabilité classique pour répondre à cette question.

Concrètement, cette théorie étudie des structures dénombrables et pose des questions de la forme « Étant donné une structure calculable $\mathcal{A}$, quels sont les degrés Turing possibles des structures isomorphes à $\mathcal{A}$? » ou bien « Étant donné deux structures calculables et isomorphes, de quelle puissance calculatoire a-t-on besoin pour calculer leur isomorphisme ? » Par exemple, les instances calculables des ordres denses sans extrémités sont toutes deux à deux calculatoirement isomorphes. On les appelle *calculatoirement catégoriques*.

Degrés d'énumération

La calculabilité classique place « le calculable » comme puissance calculatoire de référence. Mais certains problèmes s'expriment de manière naturelle sous forme d'ensembles non calculables, dont les éléments peuvent toutefois être énumérés dans le désordre par un processus calculable. On appelle ces ensembles *calculatoirement énumérables* (c. e.). En particulier, si E est un ensemble d'entiers c. e. et si $n \in E$, il est possible de s'en rendre compte en un temps fini, en lançant la procédure d'énumération et en attendant que n apparaisse. En revanche, si $n \notin E$, alors il ne sera pas possible en général de le savoir en un temps fini. Par exemple, l'ensemble des équations diophantiennes (des équations à coefficients entiers, comme $3x^3 - 2y^2 + x - 2 = 0$) qui admettent des solutions entières est calculatoirement énumérable, car il suffit de chercher exhaustivement des solutions, et d'énumérer l'équation si une telle solution existe.

Les degrés d'énumération placent « le calculatoirement énumérable » comme puissance de référence. On peut définir une *réduction d'énumération* $A \leqslant_e B$ ssi toute énumération des éléments de B calcule une énumération des éléments de A. Cette réduction est une pré-ordre partiel, qui induit une notion de *degré d'énumération* : le degré d'énumération de A est l'ensemble $\deg_e(A)$ de tous les ensembles B tels que $A \leqslant_e B$ et $B \leqslant_e A$. L'étude des degrés d'énumération munis de l'ordre partiel $\leqslant_e$ constitue une branche active de recherche en calculabilité.

Chapitre 2

Infinis de Cantor

Si l'on devait donner une date à la naissance de la logique moderne, nous situerions sans hésiter celle-ci en 1872, date à laquelle Georg Cantor expose sa première démonstration du théorème 4.1 à venir, où il établit la non-dénombrabilité des nombres réels. Cantor isolera plus tard la quintessence de cette première preuve à travers son fameux *argument diagonal*, qui aura une place centrale en calculabilité.

Georg Cantor, 1845–1918

Les travaux de Cantor sur l'infini marquent le début d'une théorie des ensembles « complexe », qui jouera un grand rôle dans la quête fondationnelle des mathématiques du début du XX^e^ siècle, dont nous parlerons en détail en première partie du chapitre 9, sur la fameuse « crise des fondements ». Cette crise débouchera sur le développement de la logique mathématique telle que nous la connaissons aujourd'hui, avec la théorie des ensembles moderne, dite ZFC, mais aussi avec le développement des premières théories du calcul, utilisées par Gödel pour montrer son fameux théorème d'incomplétude, que l'on peut considérer comme une déclinaison sophistiquée de l'argument diagonal de Cantor.