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 XX$^{\text{e}}$ 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 $R = 0.b_0b_1b_2b_3\ldots$ 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 $n_1 < n_2 < n_3 < \ldots$ tels que les tirages numéros n_1, n_2, n_3, $\ldots$ sont tous des tirages « pile ». Dans le cas où notre suite n_1, n_2, n_3, $\ldots$ 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, $\ldots$ 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.

De quoi s'agit-il exactement ? Cantor montre que les ensembles infinis n'ont pas tous la même « taille ». Il y a strictement plus de nombres réels que de nombres entiers, dans un sens que nous définirons précisément dans quelques lignes. Cantor se basera sur cette découverte pour développer une étude mathématique de l'infini. Il créera notamment les nombres transfinis, qui constitueront la colonne vertébrale des définitions mathématiques, et que nous aborderons dans le chapitre 27.

Cantor ne fut cependant pas le premier à remarquer que l'infini n'obéissait pas aux mêmes règles que le fini. En particulier, une caractéristique surprenante des ensembles infinis est que le tout n'est pas forcément plus grand que ses parties. Galilée en fait une exposition lumineuse dans son ouvrage « Discours concernant deux sciences nouvelles » [71] à travers un dialogue savoureux entre deux personnages, Salviati et Simplicio :

– Salviati. *J'estime que les épithètes comme « plus grand », « plus petit » et « égal » ne conviennent pas aux grandeurs infinies, dont il est impossible de dire que l'une est plus grande, plus petite ou égale à une autre. Mais voici pour le prouver un raisonnement qui me revient à l'esprit : vous savez parfaitement je suppose quels nombres sont carrés et quels nombres ne le sont pas.*
– Simplicio. *Je sais parfaitement qu'un nombre carré provient de la multiplication d'un autre nombre par lui-même ; ainsi quatre, neuf, etc. sont des nombres carrés résultant de la multiplication de deux, trois, etc. par eux-mêmes.*
– Salviati. *Fort bien, quant aux nombres qui ne proviennent pas de nombres multipliés par eux-mêmes, ce ne sont pas des carrés. Par conséquent, si je dis que les nombres pris dans leur totalité, en incluant les carrés et les non-carrés, sont plus nombreux que les carrés seuls, j'énoncerai, n'est-ce pas, une proposition vraie ?*
– Simplicio. *Très certainement.*
– Salviati. *Si je demande maintenant combien il y a de nombres carrés, on peut répondre sans se tromper qu'il y en a autant que de racines correspondantes, attendu que tout carré a sa racine et toute racine son carré, qu'un carré n'a pas plus d'une racine et une racine pas plus d'un carré.*
– Simplicio. *Exactement.*
– Salviati. *Mais si je demande combien il y a de racines, on ne peut nier qu'il y en a autant que de nombres, puisque tout nombre est la racine de quelque carré il faudrait admettre que les carrés sont aussi nombreux que tous les nombres pris ensemble.*

On observe à la lecture du dialogue de Galilée le piège du paradoxe se refermer sur Simplicio. Galilée utilise pour cela un concept qui sera repris par Cantor : deux ensembles A et B « ont le même nombre d'éléments » si l'on peut faire correspondre exactement les éléments de A et les éléments de B, autrement dit s'il existe une bijection entre les deux ensembles.

1. Équipotence et subpotence

Rappelons qu'une fonction $f : E \to F$ est *injective* si

$$\forall x, y \quad x \neq y \Rightarrow f(x) \neq f(y),$$

surjective si son image est l'ensemble F tout entier, et *bijective* si elle est à la fois injective et surjective.

Définition 1.1. Deux ensembles E et F sont *équipotents* s'il existe une bijection entre eux. On écrira alors $|E| = |F|$. ♢

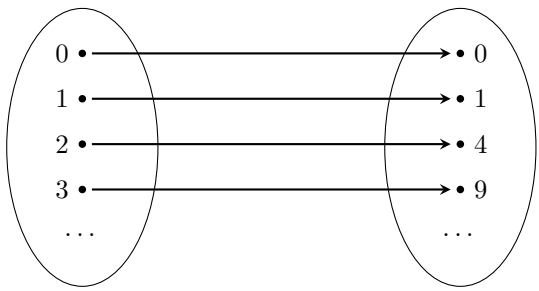

FIGURE 1.2 – *L'argument de Galilée pour dire qu'il y a « autant » d'entiers que de carrés*

Selon notre définition les entiers, et les carrés d'entiers ont donc la même cardinalité : il y a autant d'éléments dans les deux ensembles. Ce qui semble paradoxal, c'est que les carrés d'entiers forment une partie stricte de l'ensemble des entiers. Le paradoxe est résolu de la manière la plus simple qui soit : l'intuition que l'on a sur les ensembles finis, qui veut qu'une sous-partie stricte d'un ensemble contienne moins d'éléments n'est tout simplement plus vraie pour les ensembles infinis.