Calculabilité
Cours 2 : les limites du calcul

Kévin Perrot – Aix Marseille Université – printemps 2019

Table des matières

3 Les limites du calcul **1**
3.1 Enumération des machines de Turing 1
3.2 Simplifications 2
3.3 Cardinalité 2
3.4 Code d'une machine de Turing 4
3.5 Théorème de l'arrêt 5
3.6 Raisonnement par l'absurde et diagonalisation 5

3 Les limites du calcul

3.1 Enumération des machines de Turing

Remarque 1.
Enumérer *un ensemble* S = *donner (au moins) un numéro à chaque élément*
= *donner une fonction surjective de* $\mathbb{N}$ *dans* S.
Dans ce cas S *ne peut pas être plus grand que* $\mathbb{N}$.
Une énumération[1] *de* S *sans répétition (= injective) est une bijection de* $\mathbb{N}$ *dans* S.

Remarque 2. *Il y a*

$$((nm2)+1)^{(n-1)m}$$

machines de Turing à $|Q| = n$ *états et* $|\Gamma| = m$ *symboles. Il faut au moins 2 états (*q_0 *et* q_F*) et au moins 2 symboles de ruban (*B *et un autre) pour définir une MT.*

Remarque 3. *On peut écrire la définition d'une machine de Turing sur une feuille découpée en cases (ça ressemble beaucoup à un ruban). En prenant un ordre sur les symboles utilisés (ça ressemble beaucoup à un ordre sur* Γ*), on peut définir un ordre lexicographique*[2] *sur l'ensemble des machines de Turing. Donc étant donné un ensemble fini quelconque de machines de Turing, on peut les énumérer.*

Lemme 4. *Il existe une bijection entre l'ensemble des machines de Turing et* $\mathbb{N}$.

Donc il existe une **énumération de l'ensemble de toutes les machines de Turing** (et même plusieurs).

1. Il faut que l'énumération soit totale, c'est-à-dire que tout élément ait une image.
2. Comme dans un dictionnaire.

Démonstration. Nous allons énumérer sans répétition l'ensemble des machines de Turing. L'idée est de numéroter de 0 à 80 les 81 MT à 2 états et 2 symboles, puis de 81 à 2277 les 2197 MT à 2 états et 3 symboles, puis de 2278 à 30838 les 28561 MT à 3 états et 2 symboles, puis de 30339 à 47076219 les 47045881 MT à 3 états et 3 symboles, etc.

Plus formellement, soit f une bijection de $\mathbb{N}$ dans $(\mathbb{N}\setminus\{0,1\})^2$. Nous allons commencer par énumérer les MT avec $(|Q|,|\Gamma|) = f(0)$, puis les MT avec $(|Q|,|\Gamma|) = f(1)$, puis $f(2)$, etc. Pour un $(|Q|,|\Gamma|)$ donné il y a un nombre fini de machines de Turing (remarque 2), nous pouvons donc les énumérer sans problème (remarque 3). □

Remarque 5. *On peut également énumérer (c'est-à-dire mettre en bijection avec $\mathbb{N}$) les programmes en C ou python ou Java ou Hashkell* etc*, tout simplement en prenant l'ordre lexicographique sur les chaînes de caractères.*

3.2 Simplifications

Lorsque cela nous arrangera, nous pourrons nous ramener à ne considérer
— que **les langages sur** $\Sigma = \{0,1\}$ (au lieu de Σ fini quelconque),
— que **les fonctions de** $\mathbb{N}$ **dans** $\mathbb{N}$ (au lieu des fonctions de Σ^* dans Γ^*).

Remarque 6. *Quand on dit qu'une chose A « peut se ramener à » une chose B, on dit en fait plus formellement que l'ensemble des choses A peut être mis en* **bijection** *avec l'ensemble des choses B.*

Pour un Σ donné, avec une bijection de Σ^* dans $\mathbb{N}$ (par exemple à partir de l'idée de compter en base $|\Sigma|$ on peut définir $f(w_0w_1\dots w_k) = \sum_{i=0}^{k} w_i m^i$ avec $m = |\Sigma|$, où les lettres de Σ sont mises en correspondance avec $\{1,\dots,m\}$, et la convention $f(\epsilon) = 0$) on peut traduire un mot en un nombre (et réciproquement car c'est une bijection). Avec la représentation binaire de ce nombre, nous obtenons une bijection de Σ^* dans $\{0,1\}^*$.

Remarque 7.
— *Il existe une bijection entre Σ^* et $\mathbb{N}$*
— *Il existe une* **bijection entre** $\mathbb{N}\times\mathbb{N}$ **et** $\mathbb{N}$*,*
et donc également entre $\mathbb{N}^n$ et $\mathbb{N}$ pour tout $n \in \mathbb{N}$.
Par conséquent, lorsque nous parlons de fonctions calculables, nous pouvons restreindre nos considérations aux **fonctions de** $\mathbb{N}$ **dans** $\mathbb{N}$*.*

3.3 Cardinalité

Remarque 8. *Pour comparer les tailles d'ensembles infinis, la bonne notion est celle de l'existence d'une bijection. Si il existe une bijection entre deux ensembles, alors ils contiennent autant d'éléments l'un que l'autre. Il y a une excellente analogie pour se rendre compte de cela, c'est l'*hotel de Hilbert *!*

Notation 9. *L'***ensemble des parties de** $\mathbb{N}$ *(ensemble des sous-ensembles de $\mathbb{N}$) est dénoté $\mathcal{P}(\mathbb{N})$ ou $2^{\mathbb{N}}$, et est en bijection avec $[0,1]$, qui est en bijection avec $\mathbb{R}$.*

Notation 10. $|\mathbb{N}| = |\mathbb{Q}| = |\mathbb{Z}| = |\mathbb{N}\times\mathbb{N}| = \aleph_0$ *et* $|\mathbb{R}| = |\mathbb{R}\times\mathbb{R}| = |[0,1]| = 2^{\aleph_0}$.

Lemme 11. *Il existe une bijection entre l'ensemble des langages sur $\Sigma = \{0,1\}$ et $[0,1]$.*

Démonstration. L'ensemble des mots sur $\Sigma = \{0, 1\}$ est en bijection avec $\mathbb{N}$ (représentation binaire). Un langage peut donc se ramener à un sous-ensemble de $\mathbb{N}$ (un ensemble de nombres). Par conséquent l'ensemble des langages sur $\{0, 1\}$ est en bijection avec $\mathcal{P}(\mathbb{N})$, qui est en bijection avec $[0, 1]$ (notation 9). □

Lemme 12. *Il existe une bijection entre l'ensemble des fonctions de* $\mathbb{N}$ *dans* $\mathbb{N}$*, et* $[0, 1]$*.*

Démonstration. Il y a autant de fonctions de A dans B que d'éléments dans $B^{|A|}$. Dans notre cas il y a $\aleph_0^{\aleph_0}$ fonctions de $\mathbb{N}$ dans $\mathbb{N}$. Montrons que ce nombre n'est pas plus grand que $2^{\aleph_0}$, ce que nous admettrons comme suffisant pour conclure.

Pour cela nous allons construire une fonction injective de l'ensemble des fonctions de $\mathbb{N}$ dans $\mathbb{N}$, dans $[0, 1]$, ce qui suffira pour conclure. Une fonction f de $\mathbb{N}$ dans $\mathbb{N}$ est une suite infinie de nombres : $f(0), f(1), f(2)$, etc (attention : chaque $f(i)$ est un nombre fini car $+\infty \notin \mathbb{N}$). Nous pouvons donc faire correspondre à chaque fonction un nombre réel $0.f(0)f(1)f(2)\ldots$. Cette fonction n'est cependant pas injective (ni surjective). Pour la rendre injective nous pouvons utiliser le codage suivant :

— les $f(i)$ sont codés en binaire dédoublé (chaque bit est écrit deux fois, par exemple 9 en décimal devient 11000011),
— les $f(i)$ et $f(i + 1)$ sont séparés par la séquence 01.

□

Théorème 13. $\aleph_0 < 2^{\aleph_0}$**.**

Démonstration (Cantor, 1891). Nous allons démontrer ce résultat par l'absurde. Supposons qu'il existe une bijection entre $\mathbb{N}$ et $[0, 1]$ (auquel cas $\aleph_0 = 2^{\aleph_0}$), alors il est possible d'énumérer tous les nombres réels entre 0 (inclus) et 1 (exclu) sans en oublier aucun :

$$\begin{aligned} r_1 &= 0.\boxed{1}234567890\ldots \\ r_2 &= 0.5\boxed{3}49236423\ldots \\ r_3 &= 0.72\boxed{9}1655000\ldots \\ r_4 &= 0.239\boxed{3}218693\ldots \\ &\ldots \end{aligned}$$

Nous allons montrer qu'il est impossible d'avoir énuméré tous les éléments de $[0, 1]$. En effet, nous avons forcément oublié le nombre suivant :

$$r_+ = 0.2404\ldots$$

construit en prenant pour première décimale la première décimale de r_1 plus 1 modulo 10, en seconde décimale la seconde décimale de r_2 plus 1 modulo 10, en troisième décimale la troisième décimale de r_3 plus 1 modulo 10, etc, à l'infini (les nombres réels peuvent avoir une infinité de décimales). On a bien $r_+ \in [0, 1]$, et pour tout $i \in \mathbb{N} : r_i \neq r_+$ car ils diffèrent par la $i^{\text{ème}}$ décimale. □

Corollaire 14.
Il existe des langages non récursifs et des fonctions non calculables.

Démonstration. Application du théorème 13 d'après les lemmes 4 et 11 pour les langages, et lemmes 4 et 12 pour les fonctions. □

Remarque 15. *Il existe donc des infinis de tailles différentes. On dira*
— **infini dénombrable** *s'ils sont en bijection avec* $\mathbb{N}$ *(donc de taille* $\aleph_0$*),*
— **infini indénombrable** *sinon.*

Le théorème 14 (corollaire du théorème 13) ne nous donne pas d'exemple de langage non récursif / fonction non calculable, mais nous dit qu'ils / elles sont très nombreux / nombreuses (infiniment plus que les récursifs / calculables).

Le théorème 13 amène une question naturelle : existe-t-il des infinis strictement plus grands que $\aleph_0$, mais strictement plus petits que $2^{\aleph_0}$? En d'autres termes, si l'on dénote $\aleph_1$ le second plus petit infini après $\aleph_0$, est-ce que

$$2^{\aleph_0} = \aleph_1 \,?$$

Cette question fameuse est appelée **hypothèse du continu** (HC), et fut posée par Cantor. Il fallut attendre l'axiomatisation de la théorie des ensembles [3] par Zermelo et Fraenkel (ZF) au début du XX$^{\text{e}}$ siècle, une preuve par Gödel en 1938 que HC ne peut pas être réfutée dans ZF, et une preuve par Cohen en 1963 que HC ne peut pas être prouvée dans ZF, pour arriver à la conclusion suivante : HC est indépendante de ZF. Reformulé, la théorie des ensembles qui fait consensus en mathématique ne permet pas de dire si l'hypothèse du continu est vraie ou fausse, les deux éventualités sont consistantes (n'amènent pas de contradiction).

3.4 Code d'une machine de Turing

Les résultats fondamentaux sur les limites du calcul sont liés à des problèmes dans lesquels une machine de Turing doit répondre à une question sur les machines de Turing. Pour cela, il faut pouvoir donner en entrée à une machine de Turing la définition (le code, le programme) d'une autre machine de Turing. Deux possibilités :
— en donnant le numéro de la machine dans une énumération des machines de Turing,
— en écrivant le code de la machine sur la ruban.
Nous avons vu dans la section 3.1 comment énumérer les machines de Turing, voyons maintenant comment les encoder sur le ruban.

Notation 16. *Nous noterons* $\langle M \rangle$ *le* **code d'une machine de Turing**.

Il y a de nombreuses façons d'encoder les machines de Turing sur le ruban. Par exemple, en numérotant de q_1 à q_n les états et de a_1 à a_m les symboles de ruban utilisés par une machine M, et en fixant $D = 0$ et $L = 00$, il est possible d'encoder chaque transition $\delta(q_i, a_j) = (q_k, a_l, D)$ de M par la séquence

$$\text{transition} = \underbrace{0\ldots0}_{i}1\underbrace{0\ldots0}_{j}1\underbrace{0\ldots0}_{k}1\underbrace{0\ldots0}_{l}10$$

On peut alors encoder une machine complète en commençant par dire combien elle a d'états, combien elle a de symboles de ruban, puis en listant les x transitions une à une :

$$\langle M \rangle = 111\underbrace{0\ldots0}_{n}11\underbrace{0\ldots0}_{m}11\text{transition}_1 11\text{transition}_2 11\ldots 11\text{transition}_x 111.$$

3. C'est-à-dire la définition précise d'axiomes à partir desquels on dérive des théorèmes (vérités mathématiques). Avant cela (et pour Cantor notamment), on utilisait une définition intuitive (« naïve ») des ensembles. Par exemple, rien n'interdisait de considérer l'ensemble de tous les ensembles, $\mathcal{S}$. Russell souleva a ce propos un paradoxe divertissant : soit $X = \{A \in \mathcal{S} \mid A \notin A\}$, est-ce que $X \in X$?

Par convention, nous pouvons énumérer les transitions dans l'ordre lexicographique selon l'état et le symbole.

On se convaincra que le résultat suivant est vrai.

Lemme 17. *Le langage* $L_{enc} = \{w \in \{0,1\}^* \mid w = \langle M \rangle$ *pour une MT* $M\}$ *est récursif.*

3.5 Théorème de l'arrêt

Théorème 18. *La fonction* **halt** : $(\langle M \rangle, w) \mapsto \begin{cases} 0 \text{ si } M(w) \uparrow \\ 1 \text{ sinon} \end{cases}$ *n'est pas calculable.*

Démonstration. Par l'absurde, supposons qu'il existe une machine de Turing M_{halt} qui calcule la fonction *halt*. Nous pouvons alors sans difficulté construire la machine M_{diag} suivante :

$$M_{diag}(i) = \begin{cases} 1 \text{ si } M_{halt}(i,i) = 0 \\ \uparrow \text{ si } M_{halt}(i,i) = 1 \end{cases}$$

où $\uparrow$ signifie que M_{diag} entre dans une boucle infinie (et ne termine donc pas). Considérons à présent l'entrée $\langle M_{diag} \rangle$ donnée à la machine M_{diag}. Deux cas sont possibles.

- Si $M_{diag}(\langle M_{diag} \rangle) = 1$ alors, par définition de M_{diag}, nous avons $M_{halt}(\langle M_{diag} \rangle, \langle M_{diag} \rangle) = 0$ ce qui signifie, par définition de M_{halt}, que $M_{diag}(\langle M_{diag} \rangle) \uparrow$, une contradiction.
- Si $M_{diag}(\langle M_{diag} \rangle) \uparrow$ alors, par définition de M_{diag}, nous avons $M_{halt}(\langle M_{diag} \rangle, \langle M_{diag} \rangle) = 1$ ce qui signifie, par définition de M_{halt}, que $M_{diag}(\langle M_{diag} \rangle)$ s'arrête, une contradiction.

Dans les deux cas nous arrivons à une contradiction. □

3.6 Raisonnement par l'absurde et diagonalisation

N'est-il pas surprenant que des résultats si révolutionnaires admettent des preuves si courtes et simples ? On peut noter le caractère diagonal (auto-référant) des preuves de Cantor (1891) et Turing (1936). C'est également sur un argument diagonal qu'est basé le premier théorème d'incomplétude de Gödel (1931) !