Calculabilité
Cours 2 : cardinalité

Kévin Perrot – L3 Info Aix Marseille Université – printemps 2022

0 Divertissement

```
collatz(n:int)
    print n;
    si n == 1 alors stop, sinon
        si n%2 == 0 alors collatz(n/2), sinon collatz(3*n+1), finsi
    finsi
```

Est-ce que le programme `collatz` termine sur toute entrée `n` ? Comment le savoir ?
- Tester pour tout $n \leq 10^{20}$?
- Tester si $\exists$ `n,x` tel que `collatz(n)=collatz`$^{\texttt{x}}$`(n)` ?
- Vous avez un an avec 10 programmeurs surdoués. Comment faites-vous ?

Suite de Collatz (1937) : $n \mapsto \begin{cases} n/2 & \text{si } n \mod 2 == 0 \\ n*3+1 & \text{sinon} \end{cases}$

Conjectures réfutées par de grands contre-exemples :
- Conjecture d'Euler (1772). $\forall n > 2 : \forall x_1, ..., x_k, z \in \mathbb{N} : \sum_{i=1}^{k} x_i^n \neq z^n$.
  $n = 5$ (1966) : $27^5 + 84^5 + 110^5 + 133^5 = 144^5$
  $n = 4$ (1988) : $2682440^4 + 15365639^4 + 18796769^4 = 20615673^4$
  $n > 5$: inconnu.
- Conjecture de Pólya (1919). Plus de la moitié des entiers naturels inférieurs à un entier donné ont un nombre impair de facteurs premiers.
  (1980) : le plus petit contre exemple est 906 150 257.
- Conjecture de Mertens (1885) : prouvée fausse mais aucun contre exemple explicite connu.
  2006 : plus petit contre exemple compris entre 10^{14} et 1.59×10^{40}
- Nombre de Skewes (1914).
  1955 : il existe un tel nombre inférieur à $10^{10^{10^{963}}}$.
  1987 : il existe un tel nombre inférieur à 7×10^{370}.

La morale de cet example est que les ordinateurs n'ont pas réponse à tout !

Table des matières

3 Limites des programmes : cardinalité

Nous proposons dans cette section un argument général énonçant l'existence de limites à nos capacités de calcul par ordinateur. Nous posons la question suivante :

quelles fonctions de $\mathbb{N}$ dans $\mathbb{N}$ peut-on programmer ?

Remarque 1. *La restriction aux fonctions de $\mathbb{N}$ dans $\mathbb{N}$ reste en fait un cas très général, si l'on pense que toute donnée stockée sur ordinateur est une suite de bits, que l'on peut voir comme un nombre. C'est une question d'encodage.*

Soit $\mathcal{P}$ votre langage de programmation favori (C, Python, Haskell, Java, *etc*). L'ensemble des fonctions de $\mathbb{N}$ dans $\mathbb{N}$ est de cardinalité infinie (elles sont données mathématiquement), et l'ensemble des programmes en $\mathcal{P}$ est également de cardinalité infinie (ils sont données par un code source). Cependant, on sait depuis les travaux du mathématicien Cantor en 1891, qu'il existe des ensembles infinis "plus grands" que d'autres, et l'on va voir maintenant qu'il est impossible d'avoir un programme pour chaque fonction.

Remarque 2. *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, ils ont la même cardinalité. Il y a une excellente analogie pour se rendre compte de cela, c'est l'*hotel de Hilbert *!*

Lemme 3. *Il existe une bijection entre l'ensemble des programmes en $\mathcal{P}$ et $\mathbb{N}$.*

Démonstration. Bien que l'on ne connaisse pas exactement le langage $\mathcal{P}$ dans tous ses détails[1], on sait qu'un programme consiste en un fichier de texte, dont les caractères sont choisis parmi un alphabet fini : en général ASCII, 127 caractères. On peut alors mettre en correspondance les entiers naturels et les textes, en regardant chaque texte comme un nombre écrit en base 128 (on utilise pas le zéro pour bien différencier 0001 de 1 par exemple). On a ainsi un ordre pour énumérer tous les textes en partant du texte correspondant à 1, et en comptant en base 128. Ensuite à chaque fois qu'un texte correspond à un programme qui respecte la grammaire du langage $\mathcal{P}$, on lui attribue un entier naturel, en partant de 0 pour le premier programme rencontré. Chaque programme en $\mathcal{P}$ sera associé à un entier naturel unique, et puisqu'il existe des programmes toujours plus longs les uns que les autres (on peut simplement imaginer rajouter des caractères espace) alors chaque entier naturel se verra au bout d'un moment associer un programme en $\mathcal{P}$. Cette correspondance est donc bien une bijection (injective et surjective). □

1. Ce sera un très gros avantage des machines de Turing : leur définition complète tient en quelques lignes. La définition complète d'un langage de programmation est implicitement donnée par le code source d'un compilateur...

Remarque 4.
Enumérer *un ensemble* S $=$ *donner (au moins) un numéro à chaque élément*
$=$ *donner une fonction totale surjective de* $\mathbb{N}$ *dans* S.
Dans ce cas S *ne peut pas être plus grand que* $\mathbb{N}$.
Une énumération de S *sans répétition (= injective) est une bijection de* $\mathbb{N}$ *dans* S.

Rappel : $[0, 1]$ est l'ensemble des nombres réels entre 0 et 1 (inclus), et les réels peuvent avoir une infinité de décimales (comme par exemple π).

Lemme 5. *Il existe une bijection entre l'ensemble des fonctions de* $\mathbb{N}$ *dans* $\mathbb{N}$, *et* $[0, 1]$.

Démonstration. Soit F l'ensemble des fonctions de $\mathbb{N}$ dans $\mathbb{N}$. Nous allons donner une fonction injective de F dans $[0, 1]$ (pour montrer $|F| \leq |[0, 1]|$), et une fonction injective de $[0, 1]$ dans F (pour montrer $|[0, 1]| \leq |F|$). Nous pourrons alors en déduire [2] l'énoncé du lemme (c'est-à-dire $|[0, 1]| = |F|$).

Commencons par construire une fonction injective de $[0, 1]$ dans F. Soit [3] $x = 0.x_0x_1x_2x_3...$ un réel entre 0 et 1 avec $x_i \in \{0, ..., 9\}$ pour tout $i \in \mathbb{N}$, nous pouvons lui faire correspondre la fonction $f : \mathbb{N} \to \mathbb{N}$ définie par $f(i) = x_i$.

Construisons maintenant une fonction injective de F dans $[0, 1]$. Une fonction f de $\mathbb{N}$ dans $\mathbb{N}$ est une suite infinie d'entiers naturels : $f(0)$, $f(1)$, $f(2)$, 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. 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 6. $\aleph_0 < 2^{\aleph_0}$, *avec* $\aleph_0 = |\mathbb{N}|$ *et* $2^{\aleph_0} = |[0, 1]|$.

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 et 1 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 $[0, 1]$, ce qui est une conradiction. 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, pour seconde décimale la seconde décimale de r_2 plus 1 modulo 10, pour troisième

2. Cette déduction est donnée par l'application d'un résultat classique de théorie des ensemble, appelé théorème de Cantor-Schröder-Bernstein.

3. Remarquons entre parenthèse qu'un nombre réel peut avoir plusieurs représentations, comme par exemple 1 qui peut s'écrire 1.00000... ou 0.99999..., mais cela n'a pas d'influence sur notre argumentation.

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 sur leur $i^{\text{ème}}$ décimale. □

Corollaire 7.
Dans tout langage de programmation il existe des fonctions non calculables.

Démonstration. Application du théorème 6 d'après les lemmes 3 et 5. □

Remarque 8. *Il existe donc des infinis de tailles différentes. On dira qu'un ensemble est*
- **fini** *s'il contient un nombre fini d'éléments (sa taille est un entier naturel),*
- **dénombrable** *s'il est en bijection avec* $\mathbb{N}$ *(de taille* $\aleph_0$*),*
- **indénombrable** *sinon.*

Le corollaire 7 ne nous donne pas d'exemple de fonction non calculable, mais il nous dit qu'elles sont très nombreuses (infiniment plus que les fonctions calculables) !

Digression sur l'hypothèse du continu

Le théorème 6 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

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

Cette question fameuse est appelée **hypothèse du continu** (HC), et fut posée par Cantor. Il fallut attendre l'axiomatisation de la théorie des ensembles [4] par Zermelo et Fraenkel, notée ZFC, au début du XX$^{\text{e}}$ siècle, une preuve par Gödel en 1938 que HC ne peut pas être réfutée dans ZFC, et une preuve par Cohen en 1963 que HC ne peut pas être prouvée dans ZFC, pour arriver à la conclusion suivante : HC est indépendante de ZFC. Pour reformuler, la théorie des ensembles qui fait consensus comme capturant "l'ensemble des mathématiques", ne permet pas de dire si l'hypothèse du continu est vraie ou fausse, les deux éventualités sont consistantes (n'amènent pas de contradiction).

4. 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 $X \in X$? C'est à ce moment que je vous conseille la lecture de la bande dessinée *Logicomix* par Doxiadis et Papadimitriou.