T.R.I.*: De la semi-décidabilité incalculable à l'indépendance

Tarik Kaced

17 février 2009

Table des matières

1 Résumé

Mon étoile : l'indécidabilité

La question initiale est de s'intéresser à la notion d'indécidabilité et en particulier à se demander si certains indécidables le sont plus que d'autres

En particulier En cours, nous avions décidé –en cours– de nous intéresser d'une part aux systèmes d'équations polynomiales dont l'existence d'un nombre fini de solutions est indécidable et d'autre part à la Conjecture de Goldbach (**CG**). Il semblait alors qu'en les comparant, l'indécidabilité de ces deux problèmes n'était pas de nature égale.

*Travail de Recherche Individualisé: invitation ludico-éducative à l'épanouissement des étudiants théoriciens du cours de logique linéraire

2 Travail

2.1 Définitions

Pour clarifier dès le départ quelques notions, faisons quelques définitions. Celles-ci vous seront déjà familières mais le fait de les écrire aide à la compréhension de l'auteur et du lecteur.

2.1.1 Logique

Définition. *Une théorie est un ensemble d'axiomes.*

Définition. *On dit qu'une proposition est décidable si elle est prouvablement vraie ou prouvablement fausse.*

Définition. *Un proposition est dite indécidable si elle n'est pas décidable.*

Définition. *Une théorie est consistante si on ne peut pas prouver faux dedans.*

Définition. *Une proposition p est indépendante d'une théorie $\mathcal{T}$ si : « $\mathcal{T}$ est consistante » implique « $\mathcal{T} + \{p\}$ et $\mathcal{T} + \{\neg p\}$ sont consistantes »*

2.1.2 Calculabilité

Je suppose que la notion de machine de Turing est connue et que le mot calculabilité évoque de bon souvenirs (sinon voir [4]). On ne va considérer que des sous ensembles des naturels.

Définition. *Un ensemble est récursif (ou* calculable*) s'il existe une machine de Turing qui sait décider de l'appartenance d'un élément à cet ensemble.*

Définition. *Un ensemble est récursivement énumérable (ou* semi-décidable*) s'il existe une machine de Turing dont l'ensemble de définition est cet ensemble.*

Définition. *Un système de programmation acceptable est la donnée d'une énumération infiniment redondante de toutes les machines de Turing et d'une machine universelle.*

2.1.3 La hiérarchie arithmétique

La hiérarchie arithmétique est un treillis d'ensembles d'ensembles de plus en plus difficilement calculables

Définition.

- *A est $\Delta_0 (= \Pi_0 = \Sigma_0)$ ssi A est calculable*
- *A est Σ_n ss'il existe une relation $\mathcal{R}$ calculable*[1] *telle que*

$$x \in A \Leftrightarrow \exists y_1 \forall y_2 \dots \mathrm{Q} y_n,\ \mathcal{R}(x, y_1, y_2, \dots, y_n)$$

  Où Q *vaut $\exists$ si n est impair et $\forall$ sinon*
- *A est Π_n ss'il existe un relation $\mathcal{R}$ calculable telle que*

$$x \in A \Leftrightarrow \forall y_1 \exists y_2 \dots \mathrm{Q} y_n,\ \mathcal{R}(x, y_1, y_2, \dots, y_n)$$

  Où Q *vaut $\exists$ si n est pair et $\forall$ sinon*
- *A est Δ_n si A est Σ_n et Π_n*

[1] dont l'ensemble caractéristique est calculable

2.2 Conjecture de Goldbach

Je préfère énoncer cette conjecture de la façon équivalente suivante :

Énoncé en français : Tout entier naturel plus grand que l'unité est la moyenne de deux nombres premiers

Conjecture. $\forall n \in \mathbb{N}\ \exists p, q \in \mathbb{P},\ n = \frac{p+q}{2}$

Statut de cette conjecture[7]

- Considérée comme vraie par une majorité de mathématiciens
- Un théorème proche a déjà été démontré.
- Vérifiée jusqu'à 10^{14}

2.2.1 Du point de vue logique

La fausseté de **CG** signifie qu'il existe un contre-exemple. Supposons que **CG** soit fausse, alors elle est prouvablement fausse *i.e. décidable* car cela revient à dire qu'il existe un contre-exemple. Une une machine de Turing énumérant toutes les possibilités finira par découvrir ce contre-exemple.

Soit $\mathcal{T}$ une théorie non contradictoire contenant l'arithmétique. Si **CG** indécidable dans $\mathcal{T}$ donc **CG** n'est pas prouvablement fausse (il n'existe pas de contre-exemple) et donc elle est « vraie » (mais non prouvable) au même titre que dans la proposition vraie mais non prouvable définie par Gödel dans son théorème d'incomplétude.

On peut en dire un peu plus si l'on connaît assez la théorie des modèles, le lecteur intéressé se dirigera vers [5].

2.2.2 Du point de vue calculabilité

L'ensemble $\{n \in \mathbb{N} \mid \exists p, q \in \mathbb{P},\ p < 2n, q < 2n, n = \frac{p+q}{2}\}$ des entiers vérifiant **CG** est récursif. Mais ceci n'est pas très intéressant. Cela signifie seulement qu'il existe un algorithme qui vérifie qu'un entier n'est pas un contre-exemple.

Soit la relation $\mathcal{R}$ telle que $m\mathcal{R}n$ si et seulement si n vérifie **CG** ou $n > m$. $\mathcal{R}$ est calculable donc l'ensemble $A = \{m \in \mathbb{N} \mid \forall n,\ R(n, m)\}$ est dans Π_1 dans la hiérarchie arithmétique.

2.3 L'équation diophantienne de Chaitin

Chaitin est connu pour son nombre Omega [2], le nombre de la sagesse et pour sa contribution à la théorie algorithmique de l'information.

2.3.1 Complexité de Kolmogorov-Chaitin

Les programmes ainsi que les entrée seront désormais considérés comme étant des éléments de $\{0, 1\}^*$

La complexité de Kolmogorov est une fonction exprimant la complexité de son entrée en terme d'information. Pour une entrée x, elle est définie comme étant la taille, notée $|p|$, du plus petit programme p affichant x.

Définition. *Soit x une chaîne de bits,* $K(x) = \min_p\{|p|,\ p \text{ affiche } x\}$

Évidemment, pour pouvoir parler de l'exécution d'un programme (machine de Turing) il faut se placer dans un système acceptable de programmation. La machine universelle de celui-ci sert à exécuter le programme.

Définition. *Soit x une chaîne de bits,* $K(x) = \min_p\{|p|,\ U(p) = x\}$

2.3.2 Omega de Chaitin

Ω peut être vu comme la probabilité d'arrêt d'un programme quelconque d'un certain système de programmation

Dans un système de programmation acceptable à la Chaitin on ne considère que des machines de Turing dont le domaine est préfixe[2]. La machine universelle U à donc un domaine préfixe, notons le D_U (c'est l'ensemble des programmes qui s'arrêtent)

Définition. $\Omega_U = \sum_{p \in D_U} 2^{-|p|} \leq 1$

Remarque : *Bien sûr, la valeur de Ω dépend du système de programmation dans lequel on se place*

La séquence des bits d'un réel Ω est **aléatoire** dans le sens où il existe un rang à partir duquel chaque préfixe ne peut pas être compressé par plus d'une constante.

$$\exists c\ \forall n, K(\Omega_{|n}) > n - c$$

De plus, une théorie cohérente $\mathcal{T}$ ne peut prédire qu'un nombre fini de bits de Ω et pour tout x ne peut prouver $K(x) > N$ à partir d'un certain rang N. Ce que Chaitin va en fait utiliser est que le cardinal de l'ensemble propositions prouvables dans $\mathcal{T}$ de la forme « *le n^{ieme} bit de Ω est* 1 » est fini.

2.3.3 Équation diophantienne universelle

Chaitin et d'autres avant lui [1] ont réussi à exprimer une équation diophantienne exponentielle universelle. C'est à dire qu'on arrive à trouver explicitement un système de programmation acceptable à base d'équations diophantiennes exponentielles.

Remarque : *Il me semble qu'on avait parlé en cours d'équations diophantiennes normales, mais je crois qu'il existe une méthode pour transformer une équation diophantienne exponentielle en une équation diophantienne tout court*

2.3.4 La démarche exacte de Chaitin

Chaitin [3] se place dans plusieurs systèmes de programmations acceptables : les machines à registres, le Lisp*, les équations diophantiennes exponentielles.*

- Écrire un interpréteur de programme Lisp dans un langage machine (machine à registre)
- Écrire un programme en Lisp calculant un certain Ω. *Ce nombre n'est pas calculable, mais un programme peut l'approximer par en-dessous*

[2] si x est dans le domaine, aucun de ses préfixes ne le sont

- Arithmétiser le langage des machines à registres en une équation diophantienne exponentielle *i.e construire une équation dont les solutions sont les entrées sur lesquelles la machine universelle s'arrête*
- Faire une pause café
- Écrire un programme Lisp qui sur l'entrée (n, k) calcule le n^{ieme} bit de l'approximation de Ω après k étapes et s'arrête si ce bit vaut 1 ou boucle s'il vaut 0.
- Donner le programme précédent à manger à l'interpréteur en prenant soin de l'arithmétiser lui aussi.

Construction finale À partir d'un rang $k = N$ suffisamment grand, l'approximation de Ω sera correcte jusque n et pour tout $k > N$ le programme s'arrête si et seulement si le n^{ieme} bit de Ω vaut 1. Donc il existe une infinité d'équations diophantiennes, celles pour $k > N$ assez grand, donc le cardinal de l'ensemble solution est soit fini si le n^{ieme} bit vaut 0, soit infini si le n^{ieme} bit vaut 1.

Remarque : *Bien sûr, même si les équations diophantiennes peuvent former un système de programmation acceptable, il est inutilisable en pratique et la taille de l'équation diophantienne universelle finalement obtenue est d'environ* 900000 *caractères. À défaut de l'afficher dans ce rapport, voilà une équation diophantienne prise sur [8] dont les solutions strictement positives sont les entiers premiers.*

$$
\begin{aligned}
(k+2)(1 \quad & - \quad (wz+h+j-q)^2 - ((gk+2g+k+1)(h+j)+h-z)^2 \\
& - \quad (2n+p+q+z-e)\,i^2 - \left(16(k+1)^3(k+2)(n+1)^2+1-f^2\right)^2 \\
& - \quad \left(e^3(e+2)(a+1)^2+1-o^2\right)^2 - \left((a2-1)y^2+1-x^2\right)^2 \\
& - \quad \left(16r^2y^4\left(a^2-1\right)+1-u^2\right)^2 \\
& - \quad \left(\left(\left(a+u^2\left(u^2-a\right)\right)^2-1\right)(n+4dy)^2+1-(x+cu)^2\right)^2 - (n+l+v-y)^2 \\
& - \quad \left(\left(a^2-1\right)l^2+1-m2\right)2-(ai+k+1-l-i)^2 \\
& - \quad \left(p+l(a-n-1)+b\left(2an+2a-n^2-2n-2\right)-m\right)^2 \\
& - \quad \left(q+y(a-p-1)+s\left(2ap+2a-p^2-2p-2\right)-x\right)^2 \\
& - \quad \left(z+pl(a-p)+t\left(2ap-p^2-1\right)-pm\right)^2)
\end{aligned}
$$

2.3.5 Équations diophantiennes

Le dixième problème de Hilbert

Matijasevic et indépendamment Chudnovsky (alors âgé de 17 ans) ont démontré qu'on ne peut pas dire si une équation diophantienne admet une solution ou non en général. Un des théorèmes les plus forts stipule qu'un programme donnant les solutions d'une équation diophantienne finira par se tromper. Un autre théorème dit que les ensembles diophantiens, dont les éléments constituent un début de solution pour une équations diophantiennes D donnée, sont exactement les ensemble semi-décidables. Donc en fait, les ensembles diophantiens sont à l'étage Σ_1 dans la hiérarchie arithmétique.

Pour le problème qui est de savoir si une équation diophantienne admet une infinité de solution : à partir du fait précédent, on peut facilement voir que ce problème est au moins Σ_1. L'énoncé qui dit qu'une équation diophantienne possède une infinité de solution peut se

mettre de la forme : $\forall n\ \exists m > n, D(m) = 0$. On peut certainement ramener ceci à dire qu'un certain ensemble est Σ_2 ou Π_2.

2.4 Comparaison

Un énoncé au premier étage de la hiérarchie arithmétique Σ_1, (*respectivement* Π_1), est forcément faux (*respectivement vrai*) lorsqu'on le suppose indécidable car s'il était vrai (*respectivement faux*) la machine qui essaye de vérifier le $\exists$ s'arrête.

Dans le cas Σ_2 ou Π_2 (deuxième étage), une telle machine n'existe en général pas. Mais il se peut que le problème soit en fait au premier étage de la hiérarchie.

Chaitin réduit le problème de la valeur d'un bit de Ω au problème de savoir si un ensemble est de cardinalité ω. Mais il n'a pas la réduction dans l'autre sens. En clair, comme connaître le bit d'indice n de Ω détermine le nombre de solutions de l'équation qu'il construit, donc si ce problème était en fait à l'étage 1, tout tomberait à l'eau (le problème deviendrait semi-décidable). Cependant, on sait [6] que déterminer le n^{ieme} bit de Ω est un problème *au plus* Π_2 sous certaine conditions[3].

En fait, on peut généraliser ce raisonnement en donnant des oracles aux machine de Turing. C'est à dire que si l'on suppose que l'on sait résoudre le problème de l'arrêt, alors les ensembles anciennement à l'étage n sont maintenant à l'étage $n-1$ dans la tour de la hiérarchie arithmétique relativisée. C'est comme si le point de vue de la hiérarchie arithmétique passait du rez-de-chaussée au premier étage. Et on peut continuer comme cela ...

Questions

- Est-ce que le problème que Chaitin utilise n'est pas en fait au premier étage de la hiérarchie ?
- Peut-on dire quelque chose sur les problèmes indécidables du deuxième étage ?
- Qu'en est-il pour une proposition indépendante ? Existe-t-il des énoncés dans la hiérarchie arithmétique qui le soit ? Existe-t-il une machine du Turing qui s'arrête si et seulement si l'hypothèse du continu est vraie ?

Références

[1] Jones J.P Y.V Matijasevic. Register machine proof of the theorem on exponential diophantine representation of enumerables sets. *J. Symb. Log.*, 49 :818–829, 1984. 4

[2] Gregory J. Chaitin. Incompleteness theorems for random reals. *Adv. Appl. Math.*, 8 :119–146, 1987. 3

[3] Gregory J. Chaitin. *Algorithmic Information Theory, 3rd printing.* Cambridge University Press, 2003. 4

[4] Jr. Hartley Rogers. *Theory of Recursive Fonctions and Effective Computability.* MIT Press, 1987. 2

[5] John G. Kemeny. Undecidable problems of elementary number theory. *Math. Annalen*, 135 :160–169, 1958. 3

[3] la machine de Chaitin universelle utilisée pour calculer Ω est prouvablement universelle dans Peano

[6] Robert M Solovay. A version of omega in which zfc can not predict a single bit. *Centre for Discr. Math. and Theo. Comp. Sci.*, 104, 1999. 6

[7] Wikipédia. Conjecture de goldbach. http ://fr.wikipedia.org/Goldbach. 3

[8] Wikipédia. Diophantien. http ://fr.wikipedia.org/Diophantien. 5