Fondements de l'informatique
Logique, modèles, et calculs

Chapitre: Incomplétude de l'arithmétique

Cours INF412

de l'Ecole Polytechnique

Olivier Bournez

bournez@lix.polytechnique.fr

Version du 16 juillet 2023

Incomplétude de l'arithmétique

En 1931, Kurt Gödel a prouvé un résultat dont les conséquences philosophiques en science ont été révolutionnaires : il a ainsi prouvé que toute théorie suffisante pour capturer les raisonnements arithmétiques est nécessairement incomplète, c'est-à-dire telle qu'il existe des énoncés qui ne sont pas démontrables et dont la négation n'est pas non plus démontrable.

Ce théorème est largement considéré comme l'un des plus grands achèvements des mathématiques et de la logique du 20ème siècle.

Avec tous les ingrédients précédents, nous sommes en fait en position de comprendre ce théorème et d'en donner une preuve complète. C'est l'objet de ce chapitre : enfin, nous donnerons la preuve complète due à Turing. Nous ne ferons qu'évoquer la preuve de Gödel, qui permet d'en dire plus.

1 Théorie de l'arithmétique

1.1 Axiomes de Peano

La question à laquelle on s'intéresse est de tenter d'axiomatiser l'arithmétique, c'est-à-dire les propriétés des entiers.

Nous avons déjà présenté dans le chapitre 6, les axiomes de l'arithmétique de Robinson et les axiomes de Peano : on s'attend à ce que ces axiomes soient vérifiés sur les entiers, c'est-à-dire dans *le modèle standard des entiers* où l'ensemble de base est les entiers, et où l'on interprète $+$ par l'addition, $*$ par la multiplication, et $s(x)$ par $x+1$.

Autrement dit, on s'attend à ce que ces axiomes possèdent au moins un modèle : *le modèle standard des entiers*.

Étant donnée une formule close sur une signature contenant ces symboles, F est soit vraie soit fausse sur les entiers (c'est-à-dire dans le modèle standard des entiers). Appelons *théorie de l'arithmétique*, l'ensemble $Th(\mathbb{N})$ des formules closes F qui sont vraies sur les entiers.

1.2 Quelques concepts de l'arithmétique

Il est possible de prouver que de nombreux concepts de la théorie des nombres se définissent parfaitement à partir de ces axiomes.

Par exemple, on peut exprimer les concepts suivants :

— INTDIV(x, y, q, r) défini comme "q est le quotient et r le reste de la division euclidienne de x par y".
En effet, cela peut s'écrire par la formule :

$$(x = q * y + r \wedge r < y).$$

— DIV(y, x) défini comme "y divise x". En effet, cela peut s'écrire :

$$\exists q \ \text{INTDIV}(x, y, q, 0).$$

— EVEN(x) défini comme "x est pair". En effet, cela peut s'écrire :

$$\text{DIV}(2, x).$$

— ODD(x) défini comme "x est impair". En effet, cela peut s'écrire :

$$\neg \, \text{EVEN}(x).$$

— PRIME(x) défini comme "x est premier". En effet, cela peut s'écrire :

$$(x \geq 2 \wedge \forall y(\text{DIV}(y, x) \Rightarrow (y = 1 \vee y = x))).$$

— POWER$_2(x)$ défini comme "x est une puissance de 2". En effet, cela peut s'écrire :

$$\forall y((\text{DIV}(y, x) \wedge \text{PRIME}(y)) \Rightarrow y = 2).$$

1.3 La possibilité de parler des bits d'un entier

On peut aussi écrire des formules comme BIT(x, y) signifiant que "y est une puissance de 2, disons 2^k, et le $k$ème bit de la représentation binaire de l'entier x est 1".

Cela est plus subtil, mais possible. Cela s'écrit en effet :

$$(\text{POWER}_2(y) \wedge \forall q \forall r(\text{INTDIV}(x, y, q, r) \Rightarrow \text{ODD}(q))).$$

L'idée est que si y satisfait la formule, alors y est une puissance de 2, et donc en binaire il s'écrit 10^k pour un entier k. En divisant x par y, le reste de la division r sera les k bits de poids le plus faible de x, et le quotient q les autres bits de x, puisqu'on a $x = q * y + r$. En testant si q est impair, on "lit" le $k+1$ème bit de x, soit le bit correspondant au bit à 1 dans l'entier y codant cette position.

1.4 Principe de la preuve de Gödel

Kurt Gödel a prouvé le théorème d'incomplétude en construisant, pour n'importe quel système de preuve raisonnable, une formule ϕ de l'arithmétique qui affirme sa propre non-prouvabilité dans le système :

$$\phi \text{ est vraie } \Leftrightarrow \phi \text{ n'est pas prouvable.} \tag{1}$$

Tout système de preuve raisonnable est valide, et donc on doit avoir

$$\psi \text{ prouvable } \Rightarrow \psi \text{ est vrai.} \tag{2}$$

Alors ϕ doit être vraie, car sinon

$$\begin{array}{lll} \phi \text{ est fausse} & \Rightarrow \phi \text{ est prouvable.} & \text{(par (1))} \\ & \Rightarrow \phi \text{ est vraie.} & \text{(par (2))} \end{array}$$

La construction de ϕ est intéressante par elle même, car elle capture la notion d'auto-référence.

On reviendra sur la construction de Gödel.

2 Théorème d'incomplétude

2.1 Principe de la preuve de Turing

On va prouver le théorème d'incomplétude en utilisant une approche qui permet d'obtenir les principales conséquences du théorème, et qui est en fait due à Alan Turing.

Cette approche est plus simple, et surtout nous avons maintenant tous les ingrédients pour en faire une preuve complète, en utilisant les arguments de la calculabilité.

L'idée est de se convaincre que dans l'arithmétique de Peano, ainsi que dans tout système "raisonnable" de preuve pour la théorie de l'arithmétique :

Théorème 1 *1. L'ensemble des théorèmes (formules closes prouvables) à partir des axiomes de Peano (ou de toute axiomatisation "raisonnable" des entiers) est récursivement énumérable.*

2. L'ensemble $Th(\mathbb{N})$ des formules closes F vraies sur les entiers n'est pas récursivement énumérable.

Par conséquent, les deux ensembles ne peuvent pas être les mêmes, et le système de preuve ne peut pas être complet. En clair :

Corollaire 1 *Il existe donc des formules closes vraies de $Th(\mathbb{N})$ qui ne sont pas prouvables , ou dont la négation n'est pas prouvable à partir des axiomes de Peano, ou de toute axiomatisation "raisonnable" des entiers.*

C'est le premier théorème d'incomplétude de Kurt Gödel.

Exercice 1 *(corrigé page 242) Comment concilier le théorème d'incomplétude précédent (de Gödel) avec le théorème de complétude (de Gödel) ?*

2.2 Le point facile

L'ensemble des théorèmes (formules closes prouvables à partir des axiomes de Peano) de l'arithmétique est certainement récursivement énumérable : quelle que soit la méthode de preuve (par exemple celle du chapitre 6), on peut énumérer les théorèmes en énumérant les axiomes et en appliquant systématiquement toutes les règles de déduction dans toutes les façons possibles, en émettant toutes les formules closes qui peuvent être dérivées.

Cela reste vrai dès que l'on suppose que l'on peut énumérer les axiomes de l'axiomatisation dont on part. C'est pourquoi, on peut dire que l'ensemble des théorèmes de toute axiomatisation raisonnable des entiers est récursivement énumérable.

Remarque 1 *Autrement dit, plus haut, si on veut une définition de "raisonnable", il suffit de prendre "récursivement énumérable".*

2.3 Lemme crucial

Le point crucial est alors de prouver le résultat suivant.

Lemme 1 *L'ensemble $Th(\mathbb{N})$ n'est pas récursivement énumérable.*

On prouve cela en réduisant le complémentaire $\overline{\mathrm{HP}}$ du problème de l'arrêt des machines de Turing à ce problème, i.e. en montrant que $\overline{\mathrm{HP}} \leq_m Th(\mathbb{N})$.

Le théorème découle alors :

— du fait que $\overline{\mathrm{HP}}$ n'est pas récursivement énumérable ;

— et du fait que si $A \leq_m B$ et que si A n'est pas récursivement énumérable, alors B non plus.

Rappelons que L_{univ} est le problème suivant : on se donne $\langle\langle M\rangle, w\rangle$, et on veut déterminer si la machine de Turing M s'arrête sur l'entrée w.

Étant donné $\langle\langle M\rangle, w\rangle$, nous montrons comment produire une formule close γ sur la signature de l'arithmétique telle que

$$\langle\langle M\rangle, w\rangle \in \overline{\mathrm{HP}} \Leftrightarrow \gamma \in Th(\mathbb{N}).$$

En d'autres termes, étant donné M et w, on doit construire une formule close γ sur la signature de l'arithmétique qui affirme que "la machine de Turing M ne s'arrête pas sur l'entrée w".

Cela s'avère possible parce que le langage de l'arithmétique est suffisamment puissant pour parler des machines de Turing et du fait qu'elles s'arrêtent.

En utilisant le principe de la formule $\mathrm{BIT}(y,x)$ précédent, on va construire une série de formules dont le point culminant sera une formule $\mathrm{VALCOMP}_{M,w}(y)$ qui dit que y est un entier qui représente une suite de configurations de M sur l'entrée w : en d'autres termes, y représente une suite de configurations $C_0, C_1, \cdots, C_t$ de M, codée sur un certain alphabet Σ telle que :

— C_0 est la configuration initiale $C[w]$ de M sur w ;

— C_{i+1} est la configuration successeur de C_i, selon la fonction de transition δ de la machine de Turing M, pour $i < t$;

— C_t est une configuration acceptante.

Une fois que l'on a réussi à écrire la formule $\text{VALCOMP}_{M,w}(y)$, il est facile d'écrire que M ne s'arrête pas sur l'entrée x : la formule γ s'écrit

$$\neg\exists y \ \text{VALCOMP}_{M,w}(y).$$

Cela prouve la réduction, et donc termine la preuve du lemme, et donc prouve aussi le théorème, en rappellant que $\overline{\text{HP}}$ n'est pas récursivement énumérable.

2.4 Construction de la formule

Il ne reste plus qu'à donner les détails fastidieux de la construction de la formule γ à partir de M et de w. Allons-y.

Supposons que l'on encode les configurations de M sur un alphabet fini Σ, que l'on peut supposer sans perte de restriction de taille p, avec p premier.

Chaque nombre possède une unique représentation en base p : on va utiliser cette représentation en base p plutôt que la représentation binaire, pour simplifier la discussion.

Supposons que la configuration initiale de M sur $w = a_1 a_2 \cdots a_n$ soit codée par l'entier dont les chiffres de l'écriture en base p soient respectivement $q_0 a_1 a_2 \cdots a_n$: on utilise la représentation de la définition 7.4 pour représenter les configurations.

Considérons que le symbole de blanc **B** est codé par le chiffre k en base p.

Soit LEGAL l'ensemble des 6-uplets (a,b,c,d,e,f) de nombres en base p qui correspondent aux fenêtres légales pour la machine M : voir la notion de fenêtre légale du chapitre 7. Si l'on préfère, LEGAL est l'ensemble des 6-uplets (a,b,c,d,e,f) tels que si trois éléments de Σ représentés respectivement par a,b et c apparaissent consécutivement dans une configuration C_i, et si d,e,f apparaissent consécutivement aux mêmes emplacements dans la configuration C_{i+1}, alors cela est cohérent avec la fonction de transition δ de la machine de Turing M.

On définit maintenant quelques formules :

— $\text{POWER}_p(x)$: "Le nombre x est une puissance de p" : ici p est un nombre premier fixé. Cela s'écrit :

$$\forall y((\text{DIV}(y,x) \wedge \text{PRIME}(y)) \Rightarrow y = p).$$

— $\text{LENGTH}_p(v,d)$: "Le nombre d est une puissance de p qui donne (un majorant de) la longueur de v vu comme un mot sur l'alphabet Σ à p lettres". Cela s'écrit :

$$(\text{POWER}_p(d) \wedge v < d \wedge p * v \geq d).$$

— $\text{DIGIT}_p(v,K,b)$: "Le '$k$ème' chiffre de v écrit en base p est b (où $K = p^k$)". Cela s'écrit :

$$\exists u \exists a(v = a + b * K + u * p * K \wedge a < K \wedge b < p).$$

— $3\,\text{DIGIT}_p(v,K,b,c,d)$: "Les 3 chiffres consécutifs de v à la position k sont b, c et d (où $K = p^k$)". Cela s'écrit :

$$\exists u \exists a(v = a + b*K + c*p*K + d*p*p*K + u*p*p*p*K \wedge a < K \wedge b < p \wedge c < p \wedge d < p).$$

— $\mathrm{MATCH}_p(v,L,M)$: "Les 3 chiffres de v à la position ℓ sont a, b et c et correspondent aux 3 chiffres de v à la position m selon la fonction de transition δ de la machine de Turing (où $L = p^\ell$ et $M = p^m$). Cela s'écrit :

$$\bigvee_{(a,b,c,d,e,f)\in \mathrm{LEGAL}} 3\,\mathrm{DIGIT}_p(v,L,a,b,c) \wedge 3\,\mathrm{DIGIT}_p(v,M,d,e,f).$$

Remarque 2 *On note évidemment ici,* $\bigwedge_{(a,b,c,d,e,f)\in \mathrm{LEGAL}}$ *pour la conjonction pour chacun des* 6*-uplets de* LEGAL.

— $\mathrm{MOVE}_p(v,C,D)$: "la suite v décrit [1] une suite de configurations successives de M de longueur c jusqu'à d (où $C = p^c$ et $D = p^d$) : toutes les paires de suites de 3-chiffres exactement écartées de c positions dans v se correspondent selon δ". Cela s'écrit :

$$\forall y((\mathrm{POWER}_p(y) \wedge y * p * p * C < D) \Rightarrow \mathrm{MATCH}_p(v,y,y*C)).$$

— $\mathrm{START}_p(v,C)$: "la suite v débute avec la configuration initiale de M sur l'entrée $w = a_1 a_2 \cdots a_n$ auxquelles on a ajouté des blancs B jusqu'à la longueur c ($C = p^c$; $n, p^i, 0 \le i \le n$ sont des constantes fixées qui ne dépendent que de w)". Cela s'écrit :

$$\bigwedge_{i=0}^{n} \mathrm{DIGIT}_p(v,p^i,a_i) \wedge p^n < C \wedge \forall y(\mathrm{POWER}_p(y) \wedge p^n < y < C \Rightarrow \mathrm{DIGIT}_p(v,y,B)).$$

— $\mathrm{HALT}_p(v,D)$: "La suite v possède un état d'acceptation quelque part". Cela s'écrit :

$$\exists y(\mathrm{POWER}_p(y) \wedge y < D \wedge \mathrm{DIGIT}_p(v,y,q_a)).$$

— $\mathrm{VALCOMP}_{M,w}(v)$: "La suite v est un calcul de M valide sur w". Cela s'écrit :

$$\exists c \exists d\ (\mathrm{POWER}_p(c) \wedge c < d \wedge \mathrm{LENGTH}_p(v,d) \wedge \mathrm{START}_p(v,c) \wedge \mathrm{MOVE}_p(v,c,d) \wedge \mathrm{HALT}_p(v,d)).$$

— $\gamma_{M,w}$: "La machine M ne s'arrête pas sur w". Cela s'écrit :

$$\neg \exists v\ \mathrm{VALCOMP}_{M,w}(v).$$

Notre preuve est terminée.

***Exercice 1** *(corrigé page 242) Le défaut des constructions précédentes est qu'elle permettent d'affirmer qu'il existe des formules vraies mais non-prouvables, mais sans donner le moindre exemple de fonction non-prouvable.*

Utiliser les théorèmes de point fixe de la calculabilité (chapitre précédent) pour donner explicitement une formule ψ qui n'est pas prouvable.

On verra plus loin que le second théorème de Gödel permet de faire encore mieux, et de prouver que l'on peut prendre ψ comme la formule qui affirme que la théorie est cohérente.

(la solution de l'exercice précédent produisant une formue ψ dont l'interprétation n 'est pas claire).

1. On voit un tableau à deux dimensions comme un unique mot en mettant les lignes bout à bout.

3 La preuve de Gödel

Kurt Gödel a en fait prouvé son théorème d'incomplétude d'une autre façon, en construisant une formule close qui affirme sa propre non-prouvabilité. Notons $\vdash$ pour prouvable, et $\models$ pour vrai respectivement sur les entiers.

Supposons que l'on ait fixé un codage des formules par les entiers d'une façon raisonnable : si ϕ est une formule, alors $\langle\phi\rangle$ désigne son codage (un entier).

3.1 Lemme de point fixe

Voici un lemme qui a été prouvé par Gödel, et qui ressemble fort aux théorèmes de point fixe déjà évoqués dans le chapitre précédent.

Lemme 2 (Lemme de point fixe de Gödel) *Pour toute formule $\psi(x)$ avec la variable libre x, il y a une formule close τ telle que*

$$\vdash \tau \Leftrightarrow \psi(\langle\tau\rangle),$$

i.e. les formules closes τ et $\psi(\langle\tau\rangle)$ sont prouvablement équivalentes dans l'arithmétique de Peano.

Démonstration: Soit x_0 une variable fixée. L'idée est d'observer que l'on peut construire une certaine formule SUBST(x,y,z) avec les variables libres x,y,z qui affirme "le nombre z est le codage d'une formule obtenue en substituant la constante dont la valeur est x dans toutes les occurrences de la variable libre x_0 dans la formule dont le codage est y".

Par exemple, si $\phi(x_0)$ est une formule qui contient une occurrence libre de x_0, mais aucune autre variable libre, la formule SUBST$(7,\langle\phi(x_0)\rangle,312)$ est vraie si $312 = \langle\phi(7)\rangle$.

On ne donnera pas les détails de la construction de la formule SUBST, mais l'idée est simplement d'observer que cela est bien possible, en utilisant par exemple l'idée de la relation BIT(x,y).

On considère maintenant $\sigma(x)$ définie par $\forall y$ (SUBST$(x,x,y) \Rightarrow \psi(y)$), et τ définie par $\sigma(\langle\sigma(x_0)\rangle)$.

Alors τ est la solution recherchée car

$$\begin{array}{rcl}
\tau & = & \sigma(\langle\sigma(x_0)\rangle) \\
 & = & \forall y\ (\text{SUBST}(\langle\sigma(x_0)\rangle,\langle\sigma(x_0)\rangle,y) \Rightarrow \psi(y)) \\
 & \Leftrightarrow & \forall y\ y = \langle\sigma(\langle\sigma(x_0)\rangle)\rangle \Rightarrow \psi(y) \\
 & \Leftrightarrow & \forall y\ y = \langle\tau\rangle \Rightarrow \psi(y) \\
 & \Leftrightarrow & \psi(\langle\tau\rangle)
\end{array}$$

Bien entendu, on a utilisé ici des équivalences informelles, mais l'argument peut bien se formuler dans l'arithmétique de Peano. □

3.2 Arguments de Gödel

On observe maintenant que le langage de l'arithmétique est suffisamment puissant pour parler de prouvabilité en arithmétique de Peano. En particulier, il est possible de coder une suite de formules par des entiers et d'écrire une formule PROOF(x, y) qui signifie que la suite de formules dont le codage est x est une preuve de la formule dont le codage est y.

En d'autres termes, $\vdash$ PROOF($\langle\pi\rangle, \langle\psi\rangle$) $\Leftrightarrow$ π est une preuve de ψ dans l'arithmétique de Peano.

La prouvabilité dans l'arithmétique de Peano se code donc alors par la formule PROVABLE(y) définie par $\exists x$ PROOF(x, y).

Alors pour toute formule close ϕ,

$$\vdash \phi \Leftrightarrow \models PROVABLE(\langle\phi\rangle). \tag{3}$$

On a alors :

$$\vdash \phi \Leftrightarrow \vdash PROVABLE(\langle\phi\rangle). \tag{4}$$

La direction $\Rightarrow$ est vraie car si ϕ est prouvable, alors il y a une preuve π de ϕ. L'arithmétique de Peano et le système de preuve permettent d'utiliser cette preuve pour prouver ϕ (i.e. que PROOF($\langle\pi\rangle, \langle\phi\rangle$)). La direction $\Leftarrow$ découle de 3 est de la validité de la preuve dans l'arithmétique de Peano.

Utilisons alors le lemme de point fixe à la formule close $\neg$PROVABLE(x). On obtient une formule close ρ qui affirme sa propre non-prouvabilité :

$$\vdash \rho \Leftrightarrow \neg \text{ PROVABLE}(\langle\rho\rangle),$$

en d'autres termes, ρ est vraie si et seulement si elle n'est pas prouvable dans l'arithmétique de Peano.

Par validité de la preuve dans l'arithmétique de Peano, on a

$$\models \rho \Leftrightarrow \neg \text{ PROVABLE}(\langle\rho\rangle). \tag{5}$$

Alors la formule ρ doit être vraie, puisque sinon, alors

$$\begin{array}{lcll} \models \neg\rho & \Rightarrow & \text{PROVABLE}(\langle\rho\rangle) & \text{(par 5)} \\ & \Rightarrow & \vdash \rho & \text{(par 3)} \\ & \Rightarrow & \models \rho & \text{(par validité de A. de Peano)} \end{array}$$

une contradiction.

Donc $\models \rho$. Mais maintenant,

$$\begin{array}{lcll} \models \rho & \Rightarrow & \neg\text{PROVABLE}(\langle\rho\rangle) & \text{(par 5)} \\ & \Rightarrow & \not\models \rho & \text{(par définition de la vérité)} \\ & \Rightarrow & \not\vdash \rho & \text{(par 3)} \end{array}$$

Donc ρ est vraie, mais n'est pas prouvable.

3.3 Second théorème d'incomplétude de Gödel

Le défaut de la preuve précédente est bien entendu qu'elle ne donne pas un grand sens à la formule ρ.

Le second théorème d'incomplétude de Kurt Gödel donne un exemple explicite de formule non prouvable.

On peut exprimer une formule CONSIST qui exprime le fait que la théorie est cohérente. On écrit qu'il n'est pas possible de prouver une formule F et sa négation : il suffit d'écrire $\neg\exists x(\text{PROVABLE}(x) \wedge \text{PROVABLE}(y) \wedge NEG(x,y))$, où $NEG(x,y)$ signifie que y est le codage de la négation de la formule codée par x.

Le second théorème d'incomplétude de Gödel permet de prouver que cette formule précise n'est pas prouvable.

Autrement dit :

Théorème 2 (Second théorème d'incomplétude de Gödel) *Aucun système de déduction cohérent ne peut prouver sa propre cohérence.*

Nous ne rentrerons pas plus dans les détails.

4 Notes bibliographiques

Lectures conseillées Pour aller plus loin sur les notions évoquées dans ce chapitre, nous suggérons la lecture des derniers chapitres de [Kozen, 1997], qui reste courts et directs, ou de l'ouvrage [Cori and Lascar, 1993] pour une preuve complète.

Bibliographie Ce chapitre est repris des trois derniers chapitres de l'excellent livre [Kozen, 1997].

Index

Bibliographie

[Cori and Lascar, 1993] Cori, R. and Lascar, D. (1993). *Logique Mathématique, volume II*. Masson.

[Kozen, 1997] Kozen, D. (1997). *Automata and computability*. Springer Verlag.