Pour toutes formules ϕ, ψ, θ :

$$\begin{array}{rcl}
(\phi \wedge \bot) & \Leftrightarrow & \bot \\
(\phi \wedge \psi) & \Leftrightarrow & (\psi \wedge \phi) \\
(\phi \wedge \top) & \Leftrightarrow & \phi \\
(\phi \wedge (\psi \wedge \theta)) & \Leftrightarrow & ((\phi \wedge \psi) \wedge \theta) \\
(\neg\neg\phi) & \Leftrightarrow & \phi \\
(\neg\top) & \Leftrightarrow & \bot \\
(\neg(\phi \wedge \psi)) & \Leftrightarrow & ((\neg\phi) \vee (\neg\psi)) \\
(\phi \wedge (\psi \vee \theta)) & \Leftrightarrow & ((\phi \wedge \psi) \vee (\phi \wedge \theta)) \\
((\neg\phi) \vee \psi) & \Leftrightarrow & (\phi \Rightarrow \psi)
\end{array}$$

FIG. 3 – Axiomes propositionnels

Pour tout symbole de fonction f à n arguments et tout symbole de prédicat P à n arguments,

$$\begin{array}{c}
\forall x.x = x \\
\forall x, y.\ x = y \Rightarrow y = x \\
\forall x, y, z.(x = y \wedge y = z) \Rightarrow x = z \\
\forall x_1, \ldots, x_n, y_1, \ldots, y_n.(x_1 = y_1 \wedge \ldots \wedge x_n = y_n) \Rightarrow f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n) \\
\forall x_1, \ldots, x_n, y_1, \ldots, y_n.(x_1 = y_1 \wedge \ldots \wedge x_n = y_n \wedge P(x_1, \ldots, x_n)) \Rightarrow P(y_1, \ldots, y_n)
\end{array}$$

FIG. 4 – Axiomes de l'égalité (AEq)

En plus des axiomes de la figure 2, nous utiliserons, comme dans toutes les théories logiques, les équivalences Booléennes du calcul propositionnel, données dans la figure 3, les axiomes de l'égalité, donnés dans la figure 4, les axiomes sur les quantificateurs, donnés dans la figure 5.

Théorème 3.13 *Les fonctions récursives totales sont représentables dans l'arithmétique élémentaire.*

Preuve

On montre d'abord que l'addition, la multiplication, l'égalité et les fonctions initiales sont représentables.

On représente les entiers dans la logique en base 1 : on confond $n \in \mathbb{N}$ avec sa représentation $s^n(0)$. Par contre, nous distinguerons, pour éviter toute ambiguïté, les symboles de leur interprétation, cette dernière étant notée avec un indice $\mathbb{N}$.

Montrons d'abord que, pour tous entiers m, n distincts, $T \vdash \neg s^m(0) = s^n(0)$, par récurrence sur $m+n$. Si $m = 0$ ou $n = 0$, on utilise (A_1) et éventuellement les axiomes de l'égalité. Sinon, on utilise (A_2) et l'hypothèse de récurrence.

Il en résulte de ce résultat (et des axiomes de l'égalité) que, pour tous $m, n \in \mathbb{N}$,

$$m =_{\mathbb{N}} n \Leftrightarrow T \vdash s^m(0) = s^n(0)$$

Et donc que l'égalité est représentable.

Le successeur est représentable par la formule $x = s(y)$. En effet, par définition, $m =_{\mathbb{N}} n + 1 \Rightarrow T \vdash s^m(0) = s^{n+1}(0)$. Pour la récriproque, on raisonne par récurrence sur n et on utilise les axiomes $(A_1), (A_2)$.

Pour toutes formules ϕ, ψ

$\neg(\forall x.\phi) \Leftrightarrow \exists x.\neg\phi$	
$\forall x.(\phi \wedge \psi) \Leftrightarrow (\forall x.\phi) \wedge (\forall x.\psi)$	
$\forall x.(\phi \vee \psi) \Leftrightarrow (\forall x.\phi) \vee \psi$	Si x n'est pas libre dans ψ
$\forall x.\phi \Leftrightarrow \forall y.\phi\{x \mapsto y\}$	Si $\phi\{x \mapsto y\}$ est la formule ϕ dans laquelle toute les occurrences libres de x sont remplacées par y et y n'apparait ni libre ni liée dans ϕ.

FIG. 5 – Axiomes logiques

Considérons maintenant le cas de l'addition. On montre, par récurrence sur n que, pour tous $n, m, p \in \mathbb{N}$, $T \vdash s^m(0) + s^n(0) = s^p(0) \Leftrightarrow m + n =_{\mathbb{N}} p$.

Si $n = 0$, par (A_3), et la représentation dans T de $=$,

$$T \vdash s^m(0) + 0 = s^p(0) \Leftrightarrow T \vdash s^m(0) = s^p(0) \Leftrightarrow m =_{\mathbb{N}} p$$

Pour la récurrence

$T \vdash s^m(0) + s^{n+1}(0) = s^p(0)$	$\Leftrightarrow$	$T \vdash s(s^m(0) + s^n(0)) = s^p(0)$	Par A_4 et (AEq)
	$\Leftrightarrow$	$T \vdash s^m(0) + s^n(0) = s^{p-1}(0)$	Par A_1, A_2 et (AEq)
	$\Leftrightarrow$	$m + n =_{\mathbb{N}} p - 1$	Hypothèse de récurrence
	$\Leftrightarrow$	$m + n + 1 =_{\mathbb{N}} p$	

En utilisant (A_4), la représentation du successeur et la représentation de l'égalité.

De manière analogue, la multiplication est représentable par $x \times y = z$.

L'ensemble des fonctions représentables est par ailleurs clos par composition (comme dans la preuve du théorème 3.11). Concernant la clôture par minimisation, il faut représenter l'ordre sur les entiers. Plus précisément, il faut et il suffit de montrer l'existence d'une formule $\phi_<$ telle que :

$$\begin{cases} T \vdash s^m(0) < s^n(0) \Leftrightarrow m <_{\mathbb{N}} n \\ T \vdash x < s^n(0) \Rightarrow T \vdash \exists m \in \mathbb{N}, x = s^m(0) \end{cases}$$

La deuxième condition vient de la quantification universelle : il faut s'assurer qu'il n'y a pas d"'alien" parmi les objets plus petits qu'un entier ; on utilise ensuite la même formule que dans le théorème 3.11.

On représente l'ordre strict sur les entiers par

$$\phi_<(x, y) \stackrel{\text{def}}{=} \exists z.x + s(z) = y$$

On montre, par récurrence sur n, en utilisant $(A_1, A_2, A_3, A_4, A_7)$ et la représentation du successeur, de l'égalité et de l'addition que

$$T \vdash \phi_<(s^n(0), s^m(0)) \Leftrightarrow n <_{\mathbb{N}} m$$

On montre enfin par récurrence sur n, en utilisant A_1, A_2, A_3, A_4, A_7 que $T \vdash \phi_<(x, s^n(0)) \Rightarrow \exists m.T \vdash x = s^m(0)$. Mais attention : on ne peut pas utiliser le fait que toutes les variables sont de la forme $s^m(0)$ pour un certain m : il y a des interprétations non standard.

On en déduit par le théorème 3.8 que toutes les fonctions récursives sont représentables.

Corollaire 3.14 *Toute théorie qui contient l'arithmétique élémentaire est incohérente ou incomplète.*

Preuve
D'après le théorème précédent et le lemme 3.1.

Corollaire 3.15 *L'arithmétique élémentaire est indécidable.*

On obtient l'*arithmétique de Peano* en ajoutant à l'arithmétique élémentaire un ensemble récursif d'axiomes (la *récurrence*) :

$$(\phi(0) \wedge \forall x.(\phi(x) \Rightarrow \phi(s(x))) \Rightarrow \forall x.\phi(x)$$

pour toute formule ϕ contenant x comme variable libre.

Comme nous l'avons déjà mentionné (sans le prouver), dans toute théorie récursivement axiomatisable, on peut numéroter injectivement les formules et les preuves, les images de ces codages étant des ensembles récursifs. On note $\mathsf{Code}(\phi)$ le code de la formule ϕ, $\mathsf{Preuve}(n)$ le prédicat qui énonce que n code une preuve et $\mathsf{Conclusion}(n)$, l'entier qui code la dernière formule d'une preuve. La cohérence d'une théorie T récursivement axiomatisable et qui contient l'arithmétique élémentaire s'écrit donc comme une formule

$$\mathsf{Coherent}(T) \stackrel{\text{def}}{=} \forall n.\mathsf{Preuve}(n) \Rightarrow \mathsf{Conclusion}(n) \neq \mathsf{Code}(0 = 1)$$

D'après le théorème 3.13, comme le prédicat $\mathsf{Preuve}()$ et la fonction $\mathsf{Conclusion}()$ sont récursifs, $\mathsf{Coherent}(T)$ est une formule de l'arithmétique élémentaire qui exprime la cohérence de la théorie T.

Théorème 3.16 *Si T est une théorie récursivement axiomatisable et contenant l'arithmétique de Peano, alors T est incohérente ou bien $\mathsf{Coherent}(T)$ n'est pas démontrable dans T.*

Idée de la preuve : L'idée est, en bref, de formaliser la preuve du premier théorème d'incomplétude, mais cette fois dans l'arithmétique de Peano. Plus précisément, si P est le prédicat de prouvabilité :

$$P(n) \stackrel{\text{def}}{=} \exists m.\mathsf{Preuve}(m) \wedge n = \mathsf{Conclusion}(m)$$

Alors on montre que, si $\mathcal{P}$ est l'arithmétique de Peano et $\mathcal{A}$ l'arithmétique élémentaire, $\mathcal{A} \vdash \phi$ entraine que $\mathcal{P} \vdash P(\mathsf{Code}(\phi))$. Cette partie relativement longue et besogneuse est admise ici.

Ensuite, on considère la fonction $f(x)$ définie par :

$$f(x) = \begin{cases} \mathsf{Code}(\phi(x)) & \text{si } x = \mathsf{Code}(\phi) \\ 0 & \text{sinon} \end{cases}$$

f est récursive totale, donc représentable par ϕ_f dans $\mathcal{A}$:

$$\mathcal{A} \vdash \phi_f(n, m) \Leftrightarrow n =_{\mathbb{N}} f(m)$$

Soit $\mathsf{Preuve}_T(y)$ et $\mathsf{Conclusion}_T(y)$ les analogues pour la théorie T de $\mathsf{Preuve}(y)$ et $\mathsf{Conclusion}(y)$ pour $\mathcal{A}$.Il s'agit à nouveau de fonction et prédicat récursifs (pusique T est récursivement axiomatisable), donc représentables dans $\mathcal{A}$. Soit alors

$$\psi(x) \stackrel{\text{def}}{=} \exists y.\mathsf{Preuve}_T(y) \wedge \phi_f(\mathsf{Conclusion}_T(y), x)$$

$\psi(\mathsf{Code}(\neg\psi))$ affirme alors que sa négation est démontrable.

Montrons par l'absurde que, si T est cohérente, alors $T \nvdash \neg\psi(\mathsf{Code}(\neg\psi))$. Sinon, par définition, il existe un entier n qui code cette preuve. Donc

$$\mathcal{A} \vdash \mathsf{Preuve}_T(n) \wedge \mathsf{Conclusion}_T(n) = \mathsf{Code}(\neg\psi(\mathsf{Code}(\neg\psi)))$$

Par définition de f, $\mathsf{Code}(\neg\psi(\mathsf{Code}(\neg\psi))) = f(\mathsf{Code}(\neg\psi))$. Donc

$$\mathcal{A} \vdash \mathsf{Preuve}_T(n) \wedge \phi_f(\mathsf{Conclusion}_T(n), \mathsf{Code}(\neg\psi))$$

Et donc $\mathcal{A} \vdash \psi(\mathsf{Code}(\neg\psi))$. Comme T contient $\mathcal{A}$, cela contredit la cohérence de T.

(Note : on peut aussi assez facilement montrer que $T \nvdash \psi(\mathsf{Code}(\neg\psi))$, ce qui donne explicitement une formule non démontrable dans T, sous peine d'incohérence).

Maintenant, par définition de ψ, de f et représentabilité de f :

$$\mathcal{A} \vdash (\psi(\mathsf{Code}(\neg\psi)) \rightarrow \exists y.\mathsf{Preuve}_T(y) \wedge \mathsf{Conclusion}_T(y) = \mathsf{Code}(\neg\psi(\mathsf{Code}(\neg\psi))))$$

D'autre part, comme T contient $\mathcal{P}$, $\mathcal{A} \vdash \psi(\mathsf{Code}(\neg\psi))$ entraine $T \vdash \exists y.\mathsf{Preuve}(y) \wedge \mathsf{Conclusion}(y) = \mathsf{Code}(\psi(\mathsf{Code}(\neg\psi)))$. D'où

$$T \vdash (\psi(\mathsf{Code}(\neg\psi)) \rightarrow \exists y.\mathsf{Preuve}_T(y) \wedge \mathsf{Conclusion}_T(y) = \mathsf{Code}(\psi(\mathsf{Code}(\neg\psi))))$$

On en conclut que

$$T \vdash (\psi(\mathsf{Code}(\neg\psi)) \Rightarrow \neg\mathsf{Coherent}(T))$$

Mais comme $T\neg \vdash \neg\psi(\mathsf{Code}(\neg\psi))$, $T\neg \vdash \mathsf{Coherent}(T)$.

Il en résulte que la cohérence de l'arithmétique de Peano n'est pas prouvable dans l'arithmétique de Peano. Ce qui met fin au programme de Hilbert.

4 λ-calcul pur

Le λ-calcul a été introduit par A. Church en 1930. Il s'agit d'une notation minimaliste dont le centre est la définition de fonction. A. Church a montré qu'il s'agit d'un modèle de calcul "universel" : on peut y coder par exemple les entiers et toutes les fonctions récursives partielles. Par la suite, ce calcul a inspiré de nombreux langages de programmation, dans le style fonctionnel (Lisp, ML, Scheme,...)

4.1 Termes du λ-calcul

$\mathcal{X}$ est un ensemble de symboles de variables. L'ensemble des *termes* du λ-calcul est le plus petit ensemble contenant $\mathcal{X}$ et tel que :

Abstraction : Si t est un terme et $x \in \mathcal{X}$, alors $\lambda x.t$ est un terme.

Application : Si t_1 et t_2 sont des termes, alors $(t_1)t_2$ est un terme.

Les *variables libres* et les *variables liées* d'un terme sont définies comme en logique du premier ordre, le lieur λ remplaçant les quantificateurs ; en bref, x est liée dans t s'il existe une sous-expression de t de la forme $\lambda x.u$. x est libre dans t si x n'est pas dans la portée d'une expression $\lambda x.u$. Par exemple, x est libre et liée dans $(x)\lambda x.(x)x$, x est liée et pas libre dans $(\lambda x.x)\lambda x.x$. x est libre et pas liée dans $(x)\lambda y.y$.

La relation d'α-équivalence, notée $\sim_\alpha$ (ou seulement $\sim$) est définie par renommage des variables liées : $\sim_\alpha$ est la plus petite relation d'équivalence sur les termes telle que :

- $\lambda x.t \sim_\alpha \lambda y.t\{x \mapsto y\}$ si y n'apparait pas libre dans t et $t\{x \mapsto y\}$ remplace les occurrences libres de x par y dans t
- Si $t \sim_\alpha t'$ et $u \sim_\alpha u'$, alors $\lambda x.t \sim_\alpha \lambda x.t'$ et $(t)u \sim_\alpha (t')u'$.

4.2 β-réduction

La relation de β-réduction est la plus petite relation binaire $\rightarrow_\beta$ sur les termes modulo α-conversion, telle que :

- Si les variables libres de u ne sont pas liées dans t (ce qu'on peut toujours assurer par α-conversion), alors $(\lambda x.t)u \rightarrow_\beta t\{x \mapsto u\}$
- Si $t \rightarrow_\beta u$, alors $\lambda x.t \rightarrow_\beta \lambda x.u$, $(v)t \rightarrow_\beta (v)u$ et $(t)v \rightarrow_\beta (u)v$.

On note encore $\xrightarrow[\beta]{*}$ la cloture réflexive transitive de β réduction.

Voici quelques exemples de réductions.

$$((\lambda x\lambda y.(x)y)\lambda x.x)x \underset{\beta}{\rightarrow} (\lambda y(\lambda x.x)y)x \underset{\beta}{\rightarrow} (\lambda x.x)x \underset{\beta}{\rightarrow} x$$

$$(\lambda x.(x)x)\lambda x.(x)x \underset{\beta}{\rightarrow} (\lambda x.(x)x)\lambda x.(x)x$$

Un terme est en *forme normale* s'il ne peut être β-réduit (ne contient pas de β-redex).

Un terme t est *normalisable* s'il existe un terme u en forme normale telle que $t \xrightarrow[\beta]{*} u$. u est alors une forme normale de t.

4.3 Quelques grand théorèmes

Théorème 4.1 $\xrightarrow[\beta]{*}$ *est confluente :*

$$\forall t, u, v. \exists w. (t \xrightarrow[\beta]{*} u \wedge t \xrightarrow[\beta]{*} v) \Rightarrow (u \xrightarrow[\beta]{*} w \wedge v \xrightarrow[\beta]{*} w)$$

Il en résulte qu'un terme normalisable a une unique forme normale
La stratégie de *réduction normale* consiste à réduire le redex le plus à gauche.

Théorème 4.2 *Si t est normalisable, sa forme normale s'obtient par réduction normale.*

4.4 Un combinateur de point fixe

On définit le λ-terme Y_C par :

$$Y_C = \lambda f.(\lambda x.(f)(x)x)(\lambda x.(f)(x)x)$$

Pour tout terme t, $(Y_C)t =_\beta (t)(Y_C)t$: Y_C est le combinateur de point fixe de Church.
Définissions maintenant

$$Y = (\lambda x.\lambda y.(y)(x)(x)y)\lambda x.\lambda y.(y)(x)(x)y$$

Soit t un terme quelconque.

$$\begin{array}{rcl} (Y)t & \xrightarrow[\beta]{} & (\lambda y.(y)(\lambda x.\lambda y.(y)(x)(x)y)(\lambda x.\lambda y.(y)(x)(x)y)y)t \\ & \xrightarrow[\beta]{} & (t)(\lambda x.\lambda y.(y)(x)(x)y)(\lambda x.\lambda y.(y)(x)(x)y)t \end{array}$$

Et donc $(Y)t \xrightarrow[\beta]{*} (t)(Y)t$

4.5 Codage des entiers

On définit les entiers de Church comme suit :

$$\overline{n} \stackrel{\text{def}}{=} \lambda x.\lambda y.\underbrace{(x)(x)\cdots(x)}_{n} y$$

Par exemple, $M = \lambda x.\lambda y.\lambda z.(x)(y)z$ effectue la multiplication des entiers de Church :

$$((M)\overline{n})\overline{m} \xrightarrow[\beta]{*} \overline{n \times m}$$

Il faut quelques récurrences pour le démontrer, mais voici une idée de la réduction :

$$\begin{array}{rcl} ((M)\overline{n})\overline{m} & \xrightarrow[\beta]{} & (\lambda y.\lambda z.(\lambda x.\lambda y.(x)^n y)(y)z)\overline{m} \\ & \xrightarrow[\beta]{} & \lambda z.(\lambda x.\lambda y.(x)^n y)(\lambda x.\lambda y.(x)^m y)z \\ & \xrightarrow[\beta]{} & \lambda z.\lambda y.((\lambda x.\lambda y.(x)^m y)z)^n y \\ & \xrightarrow[\beta]{} & \lambda z.\lambda y.(\lambda x.(z)^m x)^n y \\ & \xrightarrow[\beta]{n} & \lambda z.\lambda y.((z)^m)^n y \\ & \sim & \overline{n \times m} \end{array}$$

Théorème 4.3 *les fonctions récursives partielles sont représentables en λ-calcul.*

Le λ-calcul est ainsi un modèle de calcul universel. Ce qui entraine l'indécidabilité de toutes les questions correspondantes. Par exemple :

Corollaire 4.4 *L'ensemble des termes normalisables n'est pas récursif.*

Corollaire 4.5 *La β équivalence est indécidable.*