Logique et décidabilité

Nous disons qu'une formule logique est *close* si elle ne possède aucune variable libre. Par exemple, la formule $\exists x \forall y\ (x + y = y)$ est close, mais $\forall y\ (x + y = x)$ ne l'est pas puisque x est libre.

Une logique $\mathcal{L}$ est (in)décidable si son problème de validité est (in)décidable:

ENTRÉE:	une formule close φ de $\mathcal{L}$
QUESTION:	φ est vraie?

Nous montrons que l'*arithmétique de Peano 2.0*, c.-à-d. $\mathsf{FO}(\mathbb{N}, +, \times, 2^x, 3^x, \ldots, =)$, est indécidable, en donnant une réduction à partir du problème d'arrêt.

Théorème 1. *Le problème d'arrêt se réduit (au sens multivoque) au problème de validité de l'arithmétique de Peano 2.0.*

Démonstration. Nous devons construire une formule $\varphi_{\mathcal{M},w}$ qui soit vraie si et seulement si la machine de Turing $\mathcal{M}$ s'arrête sur l'entrée w. Autrement dit:

$$\langle \mathcal{M}, w \rangle \in A_{\mathsf{TM}} \iff \varphi_{\mathcal{M},w} \equiv \textit{vrai}.$$

Considérons une machine de Turing $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}})$ et un mot $w \in \Sigma^*$. Posons $B := |\Gamma| + |Q| + 1$. Nous allons représenter des suites sur alphabet $\Gamma \cup Q \cup \{\#\}$ en base B. Par exemple, pour $\Gamma = \{0, 1, \sqcup\}$ et $Q = \{q_0, q_1, q_2, q_3\}$, nous pourrions choisir ce codage:

$\sqcup$	0	1	#	q_0	q_1	q_2	q_3
0	1	2	3	4	5	6	7

Ainsi, $q_0 011 \sqcup \sqcup \cdots$ serait représentée par

$$412200 \cdots = 4 \cdot B^0 + 1 \cdot B^1 + 2 \cdot B^2 + 2 \cdot B^3 + 0 \cdot B^4 + 0 \cdot B^5 + \cdots .$$

Autrement dit, nous interprétons le chiffre de gauche comme le chiffre le moins significatif en base B. Nous choisissons un tel codage de façon arbitraire, pourvu que $\sqcup$ soit représenté par 0.

Un *historique (de calcul)* est une suite $\#C_0\#C_1\#C_2\# \cdots C_k \sqcup \sqcup \cdots$ telle que C_0 est la configuration initiale de $\mathcal{M}$ sur w, et $C_i \to C_{i+1}$ pour tout $0 \leq i < k$. Tout historique se représente par un (possiblement très grand) nombre en base B. Par exemple $\#q_0 011\#1q_1 11\#10q_2 1 \sqcup\sqcup \cdots$ correspond au nombre 351223252232162.

Nous construisons une formule qui vérifie qu'un nombre représente un historique qui atteint une configuration terminale. Cette formule est construite progressivement à partir de ces « macros »:

$\exists! i\ \psi(i)$ (il existe un unique i qui satisfait ψ):

$$\exists i\ [\psi(i) \land \forall j\ (j \neq i \to \neg\psi(j))]$$

$x \div y,\ x \bmod y$:

$$\underbrace{\exists q, r}_{\text{quotient, reste}} (x = q \cdot y + r) \land (r < y)$$

$x[i]$ ($i^{\text{ème}}$ lettre de x vu comme une suite):

$$(x \div B^i) \bmod B$$

$x[i \ldots j]$ (sous-suite de x vu comme une suite):

$$(x \div B^i) \bmod B^{j-i+1}$$

config(y):

$$(y[0] = \#) \wedge [\forall i \ (i > 0) \to (y[i] \neq \#) \wedge (\exists! j \ y[j] \in Q)]$$

succ(x, i, j, k):

$$(i < j < k) \wedge \text{config}(x[i \dots j-1]) \wedge \text{config}(x[j \dots k-1]) \wedge [(x[k] = \#) \vee (\forall \ell \ (\ell \geq k) \to (x[\ell] = \sqcup))]$$

première(x, j):

$$(j > 0) \wedge \text{config}(x[0 \dots j-1]) \wedge [(x[j] = \#) \vee (\forall \ell \ (\ell \geq j) \to (x[\ell] = \sqcup))]$$

init(x):

$$\exists j \ \text{première}(x, j) \wedge \left(x[0 \dots j-1] = q_0 \cdot B^0 + \sum_{i=1}^{|w|} B^i \cdot w_i \right)$$

trans(x, y):

Une formule qui vérifie si x et y codent des configurations C et C' telles que $C \to C'$. Pour l'implémenter, il faut extraire l'état et les lettres voisines des deux configurations et vérifier chaque transition de δ à l'aide d'une grande disjonction.

historique(x):

$$\text{init}(x) \wedge [\forall i, j, k \ \text{succ}(x, i, j, k) \to \text{trans}(x[i \dots j-1], x[j \dots k-1])]$$

arrêt(x):

$$\exists i, j \ \text{config}(x[i \dots j]) \wedge [\exists k \ (i \leq k \leq j) \wedge ((x[k] = q_{\text{acc}}) \vee (x[k] = q_{\text{rej}}))]$$

$\varphi_{\mathcal{M},w}$:

$$\exists x \ \text{historique}(x) \wedge \text{arrêt}(x)$$

Notons que la description de $\varphi_{\mathcal{M},w}$ est entièrement constructive. Ainsi, la réduction est calculable, c.-à-d. qu'il existe une machine de Turing qui calcule $\varphi_{\mathcal{M},w}$ à partir d'une description de $\mathcal{M}$ et w. $\square$

Corollaire 1. *L'arithmétique de Peano 2.0 est indécidable.*

Démonstration. Rappelons que si un langage A est indécidable et $A \leq_m B$, alors B est indécidable. Ainsi, puisque le problème d'arrêt A_{TM} est indécidable, le problème de validité pour l'arithmétique de Peano 2.0 est indécidable. $\square$