Turing-calculable ⇒ récursive

Benjamin Hellouin

Théorème 1. *Soit $M = (Q, \Sigma, B, \delta, q_0, F)$ une machine de Turing calculant la fonction $f : \Sigma^* \mapsto \Sigma^* \cup \{\top\}$. Alors la fonction f est récursive.*

Définition 1. *On considère l'alphabet $\{Q \cup \Sigma\}$. On représente une configuration donnée sous la forme d'un triplet (q, m_1, m_2) où q est l'état actuel de la machine, m_1 est le mot à gauche de la tête de lecture (exclue) et m_2 le mot à droite (dessin).*

On définit $init(x) = (q_0, 0, x)$.

Partant d'une configuration donnée, on cherche à obtenir la configuration suivante. Soit $carac(q, m_1, m_2)$ le caractère sous la tête de lecture, à savoir x_1 si $m_2 = x_1 \dots x_n$, B sinon ($rec(B, (C_a)_{a\in\Sigma})$, identité sur les éléments de l'alphabet). Pour commencer, les composantes de la fonction de transition $\delta : q \times \Sigma \to (q', s, \varepsilon)$ sont récursives, puisque le nombre d'entrées est fini (si...alors successifs, et résultats constants).

On note $c = (q, m_1, m_2) \to \delta(q, carac(c)) = (etat(c), ecriture(c), direction(c))$ récursives par composition.

On note $c = (q, x_1 \dots x_n, y_1 \dots y_n)$. Alors la fonction $config - suivante$ a comme composantes les trois fonctions récursives suivantes :
- le nouvel état $etat(c)$;
- le ruban à gauche de la tête de lecture $gauche(c)$: si $direction = -1$, on renvoie $x_1 \dots x_{n-1}$, sinon on renvoie $x_1 \dots x_{n-1} s y_1$ où $s = ecriture(c)$ et éventuellement $y_1 = B$;
- inversement pour $droite(c)$.

Il reste à prouver que ces fonctions sont récursives. A inversion près, supprimer le dernier caractère d'un mot revient à $rec(C_\varepsilon, \pi_1^2)$. La seconde fonction est une concaténation des fonctions récursives π_1^1, $ecriture$ et $rec(C_B, S_a)$.

On définit ainsi les configurations successives $config = rec(Id, config - suivante)$. Il reste à définir si l'état est final, ce qui se fait aisément par $stop(q, m_1, m_2) = 0$ si q est un état final, 1 autrement. Alors la première configuration finale atteinte lors du calcul est $config{-}finale(c) = \mu(f)$ où $f : n, c \to stop(config(n, c))$. Ainsi, partant d'un mot x, la configuration finale sera $config - finale(init(x))$.

Il ne reste qu'à exprimer le résultat du calcul par $decode : (q, m_1, m_2) \to m_1.m_2$ (concaténation). Alors, par construction, $decode(config - finale(init))$ est la fonction calculée par la machine de Turing.