Calculable implique récursif

Pierron Théo

12 avril 2014

Théorème *Soit f une fonction de $\mathbb{N} \to \mathbb{N}$ calculable par machine de Turing. Alors f est μ-récursive.*

Remarque *On s'est limité aux fonctions de codomaine $\mathbb{N}$ car $g : \mathbb{N} \to \mathbb{N}^k$ est calculable (resp. récursive) ssi $\pi_i^k \circ g$ est calculable (resp. récursive) pour tout i.*

Démonstration. f est calculable par machine de Turing donc il existe une machine de Turing déterministe $M = (Q, \Sigma, \Gamma, \delta, I, F, \#)$ qui la calcule. C'est-à-dire que si 1^n est écrit sur son ruban, l'exécution de M termine et le mot écrit sur le ruban dans la configuration finale est $1^{f(n)}$. On suppose que M est à ruban bi-infini.

On va choisir un encodage de M.

- Q est identifié à $[\![0, |Q| - 1]\!]$
- $I = \{0\}$
- F est identifié à son indicatrice. Comme F est fini, 1_F est primitive récursive.
- $\#$ est codé par 0
- Γ est codé par $[\![1, |\Gamma|]\!]$. On note $k = |\Gamma| + 1$.
- δ est une fonction qui va d'un ensemble fini dans un ensemble fini donc elle est primitive récursive

Soit (g, q, d) une configuration (g est le mot écrit strictement à gauche de la tête de lecture, avant l'infinité de 0, et d celui à droite).

On représente q par l'entier associé dans le codage de Q (qu'on renote q). On représente g par l'entier dont l'écriture k-aire est g. Comme $\#$ est représenté par 0, l'ajout de $\#$ en tête de g ne change pas cet entier.

De même, on représente d par l'entier dont l'écriture k-aire est le mot d^t. De même, l'ajout de $\#$ en fin de d ne change pas cet entier.

On doit maintenant simuler l'exécution de M en utilisant des fonctions récursives.

- init : $\mathbb{N} \to \mathbb{N}^3$ renvoie la configuration initiale quand on veut calculer $f(n)$.

  $$\text{init}(n) = \left(0, 0, \frac{k^{n+1} - 1}{k - 1}\right)$$

  L'exponentiation et la division sont primitives récursives donc init aussi.
- config_suiv : $\mathbb{N}^3 \to \mathbb{N}^3$ renvoie la configuration obtenue en un pas à partir de son argument. En remarquant que $d \bmod k$ est la lettre lue par la machine, on obtient :

  $$\text{config\_suiv}(g, q, d) =$$
  $$\begin{cases} (gk + \pi_2^3(\delta(q, d \bmod k)), \pi_1^3(\delta(q, d \bmod k)), d/k) & \text{si } \pi_3^3(\delta(q, d \bmod k)) = D \\ (g/k, \pi_1^3(\delta(q, d \bmod k)), kd + \pi_2^3(\delta(q, d \bmod k))) & \text{si } \pi_3^3(\delta(q, d \bmod k)) = G \end{cases}$$

  qui est primitive récursive.

- config : $\mathbb{N}^4 \to \mathbb{N}^3$. config$(c, i)$ renvoie la configuration obtenue en i pas à partir de c. On l'obtient par récursion primitive :

$$\text{config}(x, i) = \begin{cases} x & \text{si } i = 0 \\ \text{config\_suiv}(\text{config}(x), i - 1) & \text{sinon} \end{cases}$$

- arret : $\mathbb{N}^3 \to \mathbb{N}$ renvoie le nombre de pas pour obtenir une configuration acceptante à partir de son argument :

$$\text{arret}(c) = \mu i(1_F(\pi_2^3(\text{config}(c, i))))$$

qui est μ-récursive.

- somme : $\mathbb{N} \to \mathbb{N}$ calcule la longueur de l'écriture k-aire de son argument :

$$\text{somme}(x) = \mu(i \leqslant x)(k^i > x) - 1$$

On a finalement

$$f(x) = \text{somme}(\pi_1^3(\text{config}(\text{init}(x), \text{arret}(\text{init}(x))))) + \text{somme}(\pi_3^3(\text{config}(\text{init}(x), \text{arret}(\text{init}(x)))))$$

Donc f est μ-récursive. ■