Fonctions récursives R et RP

Laure Danthony

Table des matières

1 Définitions

NOTATION On note dans ce chapitre $\mathcal{F}_k$ l'ensemble des fonctions de $\mathbb{N}^k$ dans $\mathbb{N}$. On note $\mathcal{F} = \bigcup_{n=1}^{\infty} \mathcal{F}_n$ (notion de clôture transitive).

1.1 Les fonctions récursives primitives

DÉFINITION 1 (RP)
L'ensemble des **fonctions récursives primitives** est le plus petit ensemble de $\mathcal{F}$ contenant :

1. les fonctions *projecteurs* : $\forall n, \forall i \in [\![1, n]\!], p_i^n : (x_1, \ldots, x_n) \mapsto x_i$
2. $\forall n$, la *fonction nulle* : $O^n(x_1, \ldots, x_n) = 0$
3. la fonction *successeur* : $s : x \mapsto x + 1$

Et stable par les opérations :

1. la *composition* de fonctions $Comp_q^p$: on se donne $h : \mathbb{N}^q \to \mathbb{N}$ et des $g_i : \mathbb{N}^p \to \mathbb{N}$ ($i \in [1, q]$) et on obtient
$$Comp_q^p(x_1, \ldots, x_p) = h\big(g_1(x_1, \ldots, x_p), \ldots, g_q(x_1, \ldots, x_p)\big)$$
2. la *récurrence (primitive* sur une variable : Rec^n : on se donne $g : \mathbb{N}^n \to \mathbb{N}$ et $h : \mathbb{N}^{n+2} \to \mathbb{N}$ et on associe f définie par :
$$\begin{cases} f(0, y_1, \ldots, y_n) = g(y_1, \ldots, y_n) \\ f(x+1, y_1, \ldots, y_n) = h\big(f(x, y_1, \ldots, y_n), x, y_1, \ldots, y_n\big) \end{cases}$$

PROPOSITION 1 *Les fonctions +, ∗, … sont RP*

REMARQUE 1 Il existe des fonctions qu'on aimerait "calculables" et qui ne sont pas RP (ex Ackermann). On introduit donc les fonctions calculables "à la Church-Rosser".

1.2 Les fonctions calculables à la Church-Rosser

DÉFINITION 2 (R)
L'ensemble des fonctions calculables à la Church-Rosser est le plus petit ensemble de $\mathcal{F}$:

1. contenant RP ;
2. stable par la *minimisation* $Sch\mu^n$: à une fonction $g : \mathbb{N}^{n+1} \to \mathbb{N}$, on associe $f : \mathbb{N}^n \to \mathbb{N}$ telle que $f(x_1, \ldots, x_n) = \mathrm{Inf}\{y / g(x_1, \ldots, x_n, y) = 0\}$

REMARQUE 2 On peut faire le même travail avec des fonctions de $\mathcal{A}^*$ dans $\mathcal{A}^*$, on trouvera les mêmes résultats.

PROPOSITION 2
– *L'itération de fonctions RP reste RP.*

- *RP est stable par somme et produit bornés.*
- *RP est stable par minimisation et maximisation bornés.*
- *RP est clôt par $Sch\mu$ bornés.*

1.3 Prédicats

DÉFINITION 3
Un prédicat (partie de N^k) est RP si sa fonction caractéristique χ_p est RP

PROPOSITION 3 *Toute combinaison booléenne de fonctions RP est RP.*

PROPOSITION 4
- *Les prédicats définis à l'aide de prédicats RP (et de connecteurs) sont RP.*
- *Les prédicats définis par quantification bornées ($\forall, \exists, \ldots$) à partir de prédicats RP sont RP.*

2 Résultats importants

2.1 La fonction d'Ackermann

THÉORÈME 1 *La fonction d'Ackermann est RP mais pas R.*

2.2 Equivalence avec les machines de Turing

LEMME 1 *Si $f : \mathbb{N}^k \to \mathbb{N}$ est calculable à la Church-Rosser, alors la fonction $\widetilde{f} : \{\$, 0, 1\}^* \to \{0, 1\}$ (codage binaire puis séparation par des \$) est calculable par machine de Turing.*

THÉORÈME 2 *Toute fonction Turing-calculable est Church-Rosser-calculable.*

2.3 Théorème de la forme normale

THÉORÈME 3 *Toute fonction Church-Rosser-calculable est calculée par un algorithme de la forme :*

$$\Pi_2^1\big(\mu t(P_T(t,x) = 0)\big)$$

où P est une fonction RP

COROLLAIRE 1 *On peut réécrire toute fonction calculée par un programme avec une seule boucle "while".*