Corrigé TD 6
Calculabilité

Luc Lapointe
luc.lapointe@ens-paris-saclay.fr
home.lmf.cnrs.fr/LucLapointe/

Rappels

L'ensemble des fonctions récursives primitives RP est le plus petit ensemble de fonctions $\mathbb{N}^k \to \mathbb{N}$:

- contenant $0 : \mathbb{N}^0 \to \mathbb{N}$ telle que $() \mapsto 0$,
- contenant $Z : \mathbb{N} \to \mathbb{N}$ telle que $n \mapsto 0$,
- contenant $S : \mathbb{N} \to \mathbb{N}$ telle que $n \mapsto n + 1$,
- contenant les projections $P_k^i : \mathbb{N}^k \to \mathbb{N}$ définies pour $1 \leq i \leq k$ comme

$$(n_1, ..., n_k) \mapsto n_i,$$

- stable par composition : si $g_1, ..., g_m : \mathbb{N}^k \to \mathbb{N}$ et $f : \mathbb{N}^m \to \mathbb{N}$ sont récursives primitives, alors $\mathrm{Comp}_{m(f, g_1, ..., g_m)}$ définie comme suit est récursive primitive.

$$\begin{aligned} \mathbb{N}^k &\to \mathbb{N} \\ \vec{n} &\mapsto f(g_1(\vec{n}), ..., g_m(\vec{n})) \end{aligned}$$

- stable par récursion primitive : si $b : \mathbb{N}^k \to \mathbb{N}$ et $e : \mathbb{N}^{k+2} \to \mathbb{N}$ sont récursives primitives, alors

$$\mathrm{Prim}(b, e) : \mathbb{N}^{k+1} \to \mathbb{N}$$

  définie comme suit est récursive primitive.

$$\begin{aligned} \mathrm{Prim}(\vec{m}, 0) &\coloneqq b(\vec{m}) \\ \mathrm{Prim}(\vec{m}, n + 1) &\coloneqq e(\vec{m}, n, \mathrm{Prim}(\vec{m}, n)) \end{aligned}$$

Exercice 1 - Quelques fonctions récursives primitives

Montrer que si $b \in \mathbb{N}$ et $h : \mathbb{N}^2 \to \mathbb{N}$ est récursive primitive, alors $f : \mathbb{N} \to \mathbb{N}$ définie comme suit est récursive primitive :

$$\begin{aligned} f(0) &= b \\ f(x + 1) &= h(x, f(x)) \end{aligned}$$

En déduire que les fonctions suivantes sont récursives primitives, avec la convention que 0 correspond à *Faux* et que tout autre entier correspond à *Vrai*.

- La fonction factorielle,
- Les connecteurs logiques $\land, \lor, \neg$,
- La fonction prédécesseur.

Idée de la solution

Récursion simplifiée Tout d'abord, n'importe quel $b \in \mathbb{N}$, vu comme une fonction à 0 argument, est une fonction récursive primitive, en itérant des compositions de S sur 0.

La fonction f est récursive primitive : elle correspond à la définition de Prim lorsque b a 0 argument.

Factorielle En admettant que la multiplication de deux entiers est récursive primitive : c'est la fonction f lorsque b vaut 1 et h est la multiplication.

Connecteurs logiques
- Conjonction : La multiplication entre deux entiers fonctionne.
- Disjonction : L'addition entre deux entiers fonctionne.
- Négation : Fonction f, en considérant $b = 1$ et h la fonction nulle.

Prédécesseur C'est la fonction f, avec $b = 0$ et h la projection sur la première coordonnée.

Exercice 2 - Maximisation bornée

Soit h une fonction récursive primitive. Une fonction f est obtenue par schéma de *maximisation bornée* à partir de h en étant définie par :

$$\begin{aligned} &f(\vec{x}, y) = \text{le plus grand entier } t \leq y \text{ tel que } h(\vec{x}, t) = 0 \text{ s'il existe un tel entier,} \\ &f(\vec{x}, y) = 0 \text{ sinon.} \end{aligned}$$

Montrer que l'ensemble des fonctions récursives primitives est stable par maximisation bornée.

Solution

Cette fonction peut se réécrire en :

$$\begin{aligned} &f(\vec{x}, y) = \text{si } h(\vec{x}, y) = 0 \text{ alors } y \text{ sinon } f(\vec{x}, y - 1) \\ &f(\vec{x}, 0) = 0 \end{aligned}$$

En posant $y := n + 1$, et en admettant que les fonctions suivantes sont récursives primitives :
- la fonction $\mathrm{Eq}_0(u)$ qui vaut 0 si et seulement si $u \neq 0$
- la fonction $\mathrm{If}(b, u, v)$ qui vaut u si $b > 1$ et v sinon,

alors f est récursive primitive car c'est la fonction Prim avec $b = 0$, et

$$e(\vec{x}, n, g) = \mathrm{If}(\mathrm{Eq}_0(h(\vec{x}, n + 1)), n + 1, g)$$

Une fonction g est obtenue par schéma de *minimisation bornée* à partir de h en étant définie par :

$$g(\vec{x}, y) = \text{le plus petit entier } t \leq y \text{ tel que } h(\vec{x}, t) = 0 \text{ s'il existe un tel entier}$$
$$g(\vec{x}, y) = 0 \ \text{ sinon.}$$

Montrer que l'ensemble des fonctions récursives primitives est stable par minimisation bornée.

Idée de la solution

La fonction de minimisation bornée g peut se réécrire en :

$$\begin{aligned} g(\vec{x}, y) &= \text{si } h(\vec{x}, 0) = 0 \text{ alors } 0 \\ &\qquad \text{sinon si } h(\vec{x}, y) = 0 \wedge g(\vec{x}, y - 1) = 0 \text{ alors } y \\ &\qquad \text{sinon } g(x, y - 1) \\ g(\vec{x}, 0) &= 0 \end{aligned}$$

Exercice 3 - Paires

Montrer qu'il existe $J : \mathbb{N}^2 \to \mathbb{N}$, et $K, L : \mathbb{N} \to \mathbb{N}$ telles que :

1. J, K et L sont récursives primitives,
2. J est une bijection, et
3. on a $K(J(x, y)) = x$ et $L(J(x, y)) = y$ pour tout x, y.

Solution

cf cours

Exercice 4 - Fibonacci

Montrer que la fonction définie comme suit est récursive primitive :

$$\begin{aligned} f(0) &= 1 \\ f(1) &= 1 \\ f(n + 2) &= f(n + 1) + f(n) \end{aligned}$$

Solution

On définie $g(n)$ de la manière suivante :

$$g(0) = g(1) = J(1, 1)$$
$$\forall n > 0, \quad g(n + 1) = J((K(g(n)) + L(g(n))), g(n))$$

g est récursive primitive *(prouvez-le !)*. On peut ensuite montrer par récurrence sur n que

$$\forall n \in \mathbb{N}, \quad K(g(n + 1)) = f(n + 1) \wedge L(g(n + 1)) = f(n)$$

Cette propriété a comme corollaire que $K \circ g = f$. Or $K \circ g$ est récursive primitive.

Exercice 5 - Caractérisation des fonctions RP par itération

Soient $h : \mathbb{N}^2 \to \mathbb{N}$, et g de $\mathbb{N}^k \to \mathbb{N}$. On appelle *itération de h initiée par g* la fonction $f : \mathbb{N}^{k+1} \to \mathbb{N}$ définie par :

$$f(\vec{x}, n) = \begin{cases} g(\vec{x}) & \text{si } n = 0 \\ h(f(\vec{x}, n-1), n) & \text{sinon.} \end{cases}$$

Montrer que la plus petite classe de fonctions sur les entiers qui contient les fonctions de base, les fonctions J, K, L et qui est close par itération et composition, coïncide avec l'ensemble des fonctions récursives primitives.

Solution

J'appelle Iter l'ensemble décrit.

Preuve de RP $\subseteq$ Iter : Il suffit montrer que l'opération de récursion primitive se simule avec les opérateurs à disposition. Je note $\langle x, y\rangle$ pour $J(x, y)$.

Soit $g : \mathbb{N}^k \to \mathbb{N}$ une fonction récursive primitive. Je montre par induction sur la structure de fonction récursive primitive que $g \in$ Iter.

g est une fonction de base: Par définition de Iter, $g \in$ Iter.

g est issue d'une composition de fonctions récursives primitives: Par induction structurelle, les fonctions dont elle est composée sont dans Iter. Par stabilité de Iter par la composition, $g \in$ Iter.

g est issue d'une récursion primitive: Alors $k \geq 1$ et il existe des fonctions récursives primitives $b : \mathbb{N}^{k-1} \to \mathbb{N}$ et $e : \mathbb{N}^{k+1} \to \mathbb{N}$ telles que

$$g(\vec{m}, 0) := b(\vec{m})$$
$$g(\vec{m}, n+1) := e(\vec{m}, n, g(\vec{m}, n))$$

Par hypothèse d'induction, les fonctions e et b sont dans Iter.

Pour pouvoir simuler la récursion primitive, il faut d'une manière ou d'une autre conserver l'argument $\vec{m}$ via l'itération.

Lemme (Conservation de l'argument) Si $\lambda : \vec{x} \mapsto \lambda(\vec{x})$ est une fonction dans Iter, avec $\vec{x} \stackrel{\text{def}}{=} (x_1, ..., x_k)$, alors $\lambda' : \vec{x} \mapsto \langle x_1, ...\langle x_k, \lambda(\vec{x})\rangle ...\rangle$ également.

Preuve Par récurrence sur k. Si $k = 1$, alors $\lambda' \stackrel{\text{def}}{=} J(x, \lambda(x))$ fonctionne, et est dans Iter par composition. L'hypothèse de récurrence prouve directement l'hérédité. ■

Soit b' définie à partir de b comme dans le lemme.

On définit maintenant une fonction $\tilde{e} : \mathbb{N}^2 \to \mathbb{N}$ qui a vocation à imiter e. L'idée est que le premier argument y va transporter à la fois $\vec{m}$ et le résultat de la fonction.

$$\tilde{e}(y, n) = e\big(K(y), K \circ L(y), ..., K \circ L^{k-2}(y), n, L^{k(z)}\big)$$

$$\tilde{e}'(y, n) = \langle K(y), \langle K \circ L(y), ...\langle K \circ L^{k-2}(y), \tilde{e}(y, n)\rangle ...\rangle\rangle$$

Par composition, comme e, K, L sont dans Iter, $\tilde{e}$ et $\tilde{e}'$ le sont également.

On définit finalement g' qui imite g en transportant son argument de la manière suivante :

$$g'(\vec{m}, n) = \begin{cases} b'(\vec{m}) & \text{si } n = 0 \\ \tilde{e}'(g'(\vec{m}, n-1), n) & \text{sinon.} \end{cases}$$

La fonction g' est dans Iter par itération.

Propriété

$$\forall n \in \mathbb{N}, g'(\vec{m}, n) = \langle m_1, ... \langle m_{k-1}, g(\vec{m}, n) \rangle ... \rangle$$

Preuve Par récurrence sur n. Si $n = 0$, c'est par définition de b' et du cas de base de la définition de g. Pour l'hérédité, c'est par hypothèse de récurrence, construction de $\tilde{e}'$ et cas récursif de la définition de g.

On a finalement que g est dans Iter par composition, du fait de l'égalité suivante :

$$g(\vec{m}, n) = L^{k-1} \circ g'(\vec{m}, n)$$

Ceci conclut la preuve par induction d'inclusion de RP dans Iter.

Preuve de Iter $\subseteq$ RP : Il suffit d'exprimer l'itération comme une fonction récursive primitive. C'est direct, en utilisant la stabilité par récursion primitive : il suffit d'ignorer l'argument pris en entrée, ce qui est faisable grâce aux compostions et projections.