FONDEMENTS DE LA PROGRAMMATION

MASTER 1 INFORMATIQUE 2017-2018
INSTITUT GALILÉE - UNIVERSITÉ PARIS 13

Paulin de Naurois - Domenico Ruoppolo
(d'après un cours par Virgile Mogbil et Pierre Boudes)

TD 4: Fonctions Récursives

Exercice 1. Prouver que les fonctions suivantes sont primitives récursives:

(h) $\mathsf{sous}: \ (x, y) \in \mathbb{N} \ \mapsto \ y \dot{-} x \in \mathbb{N}$ *(fonction soustraction tronquée)*

(i) $\mathsf{fact}: \ x \in \mathbb{N} \ \mapsto \ !\, x \in \mathbb{N}$ *(fonction factorielle)*

(j) $\mathsf{mod}_2: \ x \in \mathbb{N} \ \mapsto \ [x]_2 \in \mathbb{N}$ *(fonction module 2, i.e. reste de la division par 2)*

Solution.

(h) $\mathsf{sous} = \mathsf{PR}\big(\,\mathsf{COMP}\big(\mathsf{pred}, \pi^3_3\big)\,, \mathsf{id}\,\big)$

(i) $\mathsf{fact} = \mathsf{PR}\big(\,\mathsf{COMP}\big(\mathsf{mul}, \pi^2_1, \pi^2_2\big)\,, \underline{1}\,\big)$

(j) On a

$$\begin{cases} \mathsf{mod}_2(0) & = & 0 \\ \mathsf{mod}_2(s(x)) & = & 1 \quad \text{si} \quad \mathsf{mod}_2(x) = 0 \\ \mathsf{mod}_2(s(x)) & = & 0 \quad \text{si} \quad \mathsf{mod}_2(x) = 1 \end{cases}$$

Il nous convient de définir d'abord une fonction $\mathsf{not} : \mathbb{N} \to \mathbb{N}$ qui envoie 0 en 1 e tout $n \geq 1$ en 0, elle aussi donné par récurrence primitive:

$$\begin{cases} \mathsf{not}(0) & = & 1 \\ \mathsf{not}(s(x)) & = & 0 \end{cases}$$

en sort que l'on puisse définir mod_2 simplement comme:

$$\begin{cases} \mathsf{mod}_2(0) & = & 0 \\ \mathsf{mod}_2(s(x)) & = & \mathsf{not}(\mathsf{mod}_2(x)) \end{cases}$$

Donc on définie $\mathsf{not} = \mathsf{PR}\big(\mathsf{COMP}\big(\underline{0}, \pi^2_1\big)\,, \underline{1}\,\big)$ et après $\mathsf{mod}_2 = \mathsf{PR}\big(\,\mathsf{COMP}\big(\mathsf{not}, \pi^2_2\big)\,, \underline{0}\,\big)$.

Exercice 3. Prouver que si $f : \mathbb{N} \to \mathbb{N}$ et $f' : \mathbb{N} \to \mathbb{N}$ sont primitives récursives alors la fonction $f + f' : \mathbb{N} \to \mathbb{N}$, définie par $(f + f')(x) := f(x) + f'(x)$, est primitive récursive.

Solution. $f + f' = \mathsf{COMP}\big(\mathsf{add}, f, f'\big)$

Exercice 4. Prouver que la fonction $\mathsf{min_k} : \mathbb{N}^k \to \mathbb{N}$ qui renvoie le plus petit entre k entiers naturels est récursive. Est-elle également primitive récursive?

Solution. Je veux utiliser la règle μ de Kleene. J'ai donc besoin d'une fonction récursive f telle que $f(y, x_1, \ldots, x_k) = 0 \overset{(\bullet)}{\iff} y = \min(x_1, \ldots, x_k)$, car par définition $\mu[f](x_1, \ldots, x_k) = y$ exactement lorsque $f(y, x_1, \ldots, x_k) = 0$, donc par l'équivalence $(\bullet)$ exactement lorsque $y = \min(x_1, \ldots, x_k)$. Une façon de définir telle f est de remarquer que

$$\begin{array}{rcl} y = \min(x_1, \ldots, x_k) & \iff & (y \leq x_1) \; et \; (y \leq x_2) \; et \; \ldots \; et \; (y \leq x_k) \\ & \iff & \mathsf{ppe}(y, x_1) \cdot \mathsf{ppe}(y, x_2) \cdot \ \ldots \ \cdot \mathsf{ppe}(y, x_k) = 1 \end{array}$$

où la primitive récursive ppe est celle de l'exercice 2(b). Alors la fonction f est donnée par récursion primitive comme

$$\begin{cases} f(0, x_1, \ldots, x_k) & = & \underline{0} \\ f(s(y), x_1, \ldots, x_k) & = & \mathsf{not}\big(\mathsf{ppe}(y, x_1) \cdot \mathsf{ppe}(y, x_2) \cdot \ \ldots \ \cdot \mathsf{ppe}(y, x_k)\big) \end{cases}$$

(voir la solution de l'exercice 1 pour la définition de not). Plus formellement

$$f \;=\; \mathsf{PR}\Big(\; \mathsf{COMP}\big(\mathsf{not}, \mathsf{COMP}\big(\mathsf{mul_k}, \mathsf{COMP}\big(\mathsf{ppe}, \pi_1^{k+1}, \pi_2^{k+1}\big), \ldots, \mathsf{COMP}\big(\mathsf{ppe}, \pi_1^{k+1} \pi_k^{k+1} {+1}\big)\big)\big) \;\; , \; \underline{0} \Big)$$

où $\mathsf{mul_k}$ est une version de mul avec k entrées à la place de 2 (sinon imbriquer k fois mul ci-dessus). Enfin on prend

$$\mathsf{min_k} = \mu[f]$$

On peut également éviter d'utiliser la règle μ, donc définir $\mathsf{min_k}$ directement en tant que primitive. Mais l'algorithme est plus compliqué...