# Logique et calculabilité
# TD 1 : fonctions récursives et $\lambda$-calcul

September 12, 2007

## 1 Quelques fonctions primitives récursives

**Question 1.1** *Montrez que les fonctions suivantes sont primitives récursives :*

1. $\lambda n.n \ mod \ 2$
2. $\lambda n.\lfloor \frac{n}{2} \rfloor$
3. $\lambda(\overline{n}, m).\prod_{i=0}^{m} g(\overline{n}, i)$, *où* $g$ *est une fonction primitive récursive*
4. $if\_then\_else = \lambda(c, x, y).\left| \begin{array}{ll} x & si\ c = 1 \\ y & si\ c = 0 \\ ce\ que\ vous\ voulez & sinon \end{array} \right.$
5. $\lambda n. \max_{0 \leq i \leq n} f(i)$, *où* $f : \mathbb{N} \to \mathbb{N}$ *est une fonction primitive récursive*
6. $\lambda(\overline{n}, m).\left| \begin{array}{ll} 1 & si\ \exists i \leq m \quad q(\overline{n}, i) \\ 0 & sinon \end{array} \right.$, *où* $q$ *est un prédicat primitif récursif*
7. *la fonction qui à des entiers* $n$ *et* $k$ *associe la partie entière de* $\frac{n}{2^k}$

## 2 Entiers de Church

Dans cet exercice nous allons construire un codage des entiers en $\lambda$-calcul. Ce codage sera utile pour démontrer l'équivalence entre les fonctions récursives et les termes.

**Question 2.1** *Réduire les termes suivants :*

*1.* $((\lambda x.\lambda y.(xy))\,b)\,c$

*2.* $(\lambda x.(a\,(\lambda y.(xy)))\,b)\,c$

On définit la suite de termes suivante :

$$
\begin{aligned}
\overline{0} &= \lambda f.\lambda x.x \\
\overline{1} &= \lambda f.\lambda x.(fx) \\
&\dots \\
\overline{n} &= \lambda f.\lambda x.\underbrace{(f(f\dots(f(f}_{n \text{ fois}}\,x))\dots))
\end{aligned}
$$

**Question 2.2** *Trouver un terme $\sigma$ tel que $\sigma\overline{n} \rightarrow^*_\beta \overline{n+1}$.*

**Question 2.3** *Trouver un terme $ADD$ tel que $((ADD\overline{n})\overline{m}) \rightarrow^*_\beta \overline{n+m}$.*

**Question 2.4** *Trouver un terme $MULT$ tel que $((MULT\overline{n})\overline{m}) \rightarrow^*_\beta \overline{n\times m}$.*

# 3 Codage des suites finies

Le but de cet exercice est d'associer à toute suite finie d'entiers $s \in \mathbb{N}^*$ un code numérique $\widehat{s}$ de façon à ce que les opérations usuelles sur les listes soient primitives récursives. On note $t :: q$ la liste dont la tête est $t$ et la queue $q$ ; $[]$ la liste vide. Ainsi, la liste $(a_0, \dots, a_n)$ est notée $a_0 :: \dots :: a_n :: []$

**Question 3.1** *Exhibez une bijection $c_2$ de $\mathbb{N}\times\mathbb{N}$ vers $\mathbb{N}$ vérifiant*

- *$c_2$ est primitive récursive*
- $\forall n, m \in \mathbb{N}^2,\ n \le c_2(n,m)$
- $\forall n, m \in \mathbb{N}^2,\ m \le c_2(n,m)$

**Question 3.2** *Trouvez deux fonction primitives récursives $p_2^1$ et $p_2^2$ de $\mathbb{N} \to \mathbb{N}$ telles que*

- $\forall n, m \in \mathbb{N}\times\mathbb{N} \quad p_2^1(c_2(n,m)) = n$ *et*
- $\forall n, m \in \mathbb{N}\times\mathbb{N} \quad p_2^2(c_2(n,m)) = m.$

$p_2^1$, $p_2^2$ *sont les* fonctions de projection *associées à $c_2$.*

**Question 3.3** *Déduisez-en, pour tout entier $k$ non nul, une fonction primitive récursive $c_k : \mathbb{N}^k \to \mathbb{N}$ bijective et les $k$ fonctions de projection associées.*

**Question 3.4** *La suite de Fibonacci définie par*

$$Fib(n) = \left| \begin{array}{ll} 1 & si\ n = 0\ ou\ n = 1 \\ Fib(n-1) + Fib(n-2) & sinon \end{array} \right.$$

*est-elle primitive récursive ?*

**Question 3.5** *Définissez une bijection $\widehat{\cdot} : \mathbb{N}^* \to \mathbb{N}$, qui associe à toute suite finie d'entiers $s$ un codage numérique $\widehat{s}$, telle que les fonctions suivantes soient primitives récursives :*

- *cons* $: \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *telle que* $cons(t, \widehat{q}) = \widehat{t :: q}$
- *une constante nil* $\in \mathbb{N}$ *telle que nil* $= \widehat{[]}$
- tête : $\mathbb{N} \to \mathbb{N}$ *telle que* tête$(\widehat{t :: q}) = t$
- queue : $\mathbb{N} \to \mathbb{N}$ *telle que* queue$(\widehat{t :: q}) = \widehat{q}$

**Question 3.6** *Montrez qu'avec ce même codage, les fonctions suivantes sont aussi primitives récursives :*

- longueur : $\mathbb{N} \to \mathbb{N}$ *telle que* longueur$(\widehat{a_0 :: \ldots :: a_n :: []}) = n$
- élément : $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *telle que* $\forall i \leq n$ élément$(i, \widehat{a_0 :: \ldots :: a_n :: []}) = a_i$
- appartient : $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *telle que* appartient$(n, \widehat{s})$ *teste si l'entier $n$ apparaît dans la suite codée par $\widehat{s}$*

**Question 3.7** *Soit $x_1, \ldots x_l : \mathbb{N} \to \mathbb{N}$ des fonctions primitives récursives telles que $\forall i \leq l \forall n \in \mathbb{N} \quad 0 \leq x_i(n) < n$. Soit $g : \mathbb{N}^k \to \mathbb{N}$ et $h : \mathbb{N}^{k+l+1} \to \mathbb{N}$ deux fonctions PR. Montrez que la fonction $f : \mathbb{N}^{k+1} \to \mathbb{N}$ définie par*

$$\begin{array}{rcl} f(\overline{n}, 0) & = & g(\overline{n}) \\ f(\overline{n}, m+1) & = & h(\overline{n}, m, f(\overline{n}, x_1(m+1)), \ldots, f(\overline{n}, x_l(m+1))) \end{array}$$

*est primitive récursive.*