USTHB - Faculté d'Informatique
Licence 1 2023/2024 Semestre 2
Module OPM
Date : 15/05/2024 <8h30-10h00>

Examen Final

Exercice 1 (6,5 points = 2,5+2,5+1,5) (Fonctions Récursives)

1) En utilisant la règle de récursion, montrer que la fonction F1 est Primitive Récursive :
$$F1(x,y) = |x-0| + \ldots + |x-y| = \sum_{k=0}^{y} Abs(x,k)$$
2) En déduire, en utilisant la règle de composition, que la fonction F2 est Primitive Récursive :
$$F2(x,y,z) = \sum_{k=0}^{y} |x-k+z|$$
3) Montrer que les fonctions S_n d'arité n≥1 sont Primitives Récursives :
$$S_n = \lambda x_1 \ldots x_n \,.\, (x_1+\ldots+x_n) \bmod 2$$

Indication : les fonctions suivantes sont PR : ⊕, ⊗, Moins, Fact, Abs, Exp, Sg, $\overline{Sg}$, Pred, mod, div, les fonctions constantes d'arité 0, les fonctions constantes d'arité 1.

Exercice 2 (6,5 points = (4*0,5+1) +(1,5+2)) (Machines de Turing)

1) Soit la Machine de Turing MT=<S,E,I> avec :
   I = { (1) $\mathbf{q_0 1 D q_1}$, (2) $\mathbf{q_1 1 D q_2}$, (3) $\mathbf{q_2 1 D q_1}$, (4) $\mathbf{q_1 0 G q_3}$, (5) $\mathbf{q_2 0 1 q_3}$, (6) $\mathbf{q_3 1 G q_3}$, (7) $\mathbf{q_3 0 D q_f}$ }
   a) Dérouler la machine pour x = 0, pour x = 1, pour x = 2 et pour x = 3, c'est-à-dire sur chacune des configurations initiales $q_0\underline{0}$, $q_0\underline{1}$, $q_0\underline{2}$ et $q_0\underline{3}$.
   b) Que fait la machine ?
2) On veut concevoir une machine de Turing calculant la fonction f = λx . 2x+1 :
   a) Expliquer la démarche à suivre.
   b) Donner l'ensemble I des instructions de la machine.

Exercice 3 (7 points = (1+2*1,5+1)+2) (Langage CAML)

1) On considère la fonction CAML suivante :
   let rec f x y z = if y = 1 then x :: z else hd z :: f x (y-1) (tl z) ;;
   a) Donner le type de f inféré par CAML.
   b) Dérouler la fonction f pour x = 5, y = 3 et z = [3 ;3 ;8 ;9 ;1].
   c) Dérouler la fonction f pour x = 9, y = 4 et z = [7 ;8 ;2].
   d) Que fait f ?
2) Ecrire une fonction CAML *coefficient_binomial* calculant le coefficient binomial C_i^j, pour tous entiers naturels i et j (un coefficient binomial C_i^j non nul représente l'entrée [i,j] du triangle de Pascal, avec les lignes et les colonnes numérotées à partir de 0) :

$$C_i^j = \begin{cases} 0 \;si\; j > i, \\ 1 \;si\; j = 0, \\ C_{i-1}^{j-1} + C_{i-1}^{j} \;sinon \end{cases}$$

(Illustration :
$$\begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 3 & 3 & 1 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 & 0 \\ 1 & 5 & 10 & 10 & 5 & 1 \end{matrix}$$
)

...Bon courage...

Corrigé Final OPM
Amani - Behloul - Isli

Exercice 1

1) Règle de récursion

$F1(x,0) = x = P_1^1(x)$

$F1(x,y+1) = F1(x,y) + |x-y-1|$ 2,5

$= \oplus\big(F1(x,y), Abs(x, S(y))\big)$

$= \big[\oplus\circ\big(P_2^3, Abs\circ(P_1^3, S\circ P_3^3)\big)\big](x, F1(x,y), y)$

2) Règle de composition: 2,5

$F2(x,y,z) = \sum_{k=0}^{y} |x-k+z| = F1\big(\oplus(x,z), y\big)$ PR

$= \big[F1\circ\big(\oplus\circ(P_1^3, P_3^3), P_2^3\big)\big](x,y,z)$

3) Montrons d'abord que la fonction $\Sigma_n = \lambda x_1 \ldots x_n . x_1 + \ldots + x_n$ est PR par récurrence sur $n \geq 1$

. $n=1$, $\Sigma_1 = \lambda x_1 . x_1$ donc $\Sigma_1 = P_1^1$ PR

. Supposons que Σ_n est PR 1

On a $\Sigma_{n+1}(x_1, \ldots, x_n, x_{n+1}) = \Sigma_n(x_1, \ldots, x_n) + x_{n+1}$ PR

$= \oplus\circ\big(\Sigma_n\circ(P_1^{n+1}, \ldots, P_n^{n+1}), P_{n+1}^{n+1}\big)(x_1, \ldots, x_n, x_{n+1})$

Conclusion Σ_n est PR pour tout $n \geq 1$.

La règle de composition:

Notons $r2 = \lambda x.\ x \bmod 2$ PR

0,5

alors $S_n(x_1, \ldots x_n) = [r2 \circ \Sigma_n](x_1, \ldots, x_n)$ PR

Exercice 2 :

1)

a) Déroulements :

$$q_0\underline{0} = q_0 1 \xrightarrow{(1)} 1q_1 0 \xrightarrow{(4)} q_3 1 \xrightarrow{(6)} q_3 01 \xrightarrow{(7)} q_f 1 = q_0 \underline{0}$$ 0,5

$$q_0\underline{1} = q_0 11 \xrightarrow{(1)} 1q_1 1 \xrightarrow{(2)} 11q_2 0 \xrightarrow{(5)} 11q_3 1 \xrightarrow{3\times(6)} q_3 0111$$
$$\xrightarrow{(7)} q_f 111 = q_f \underline{2}$$ 0,5

$$q_0\underline{2} = q_0 111 \xrightarrow{(1)} 1q_1 11 \xrightarrow{(2)} 11q_2 1 \xrightarrow{(3)} 111q_1 0 \xrightarrow{(4)} 11q_3 1$$
$$\xrightarrow{3\times(6)} q_3 0111 \xrightarrow{(7)} q_f 111 = q_f \underline{2}$$ 0,5

$$q_0\underline{3} = q_0 1111 \xrightarrow{(1)} 1q_1 111 \xrightarrow{(2)} 11q_2 11 \xrightarrow{(3)} 111q_1 1$$
$$\xrightarrow{(2)} 1111q_2 0 \xrightarrow{(5)} 1111q_3 1 \xrightarrow{5\times(6)} q_3 011111$$
$$\xrightarrow{(7)} q_f 11111 = q_f \underline{4}$$ 0,5

b) La machine calcule la fonction

$$f = \lambda x . \begin{cases} x & \text{si } x \text{ pair,} \\ x+1 & \text{sinon} \end{cases}$$ 1

2)

a) Démarche à suivre : (1,5)

On commence par ajouter une étoile à la fin de la représentation de x. Ensuite on réitère le processus consistant à supprimer une barre à gauche de l'étoile et à ajouter deux barres à droite de l'étoile, jusqu'à épuisement des barres à gauche de l'étoile. A la fin, on supprime l'étoile et on se positionne sur la barre la plus à gauche.

b)

$I = \{$

(1) $q_0 \,|\, D\, q_0 \longrightarrow$ (2) $q_0\, 0 * q_1 \longrightarrow$ (3) $q_1 * G\, q_1 \longrightarrow$ (4) $q_1 \,|\, G\, q_1$

$\downarrow$ (5) $q_1\, 0\, D\, q_2$

(7) $q_7\, 0\, D\, q_f \longleftarrow$ (6) $q_2 * 0\, q_7 \longleftarrow$ (5) $q_1\, 0\, D\, q_2$

$\downarrow$ (8) $q_2 \,|\, 0\, q_3$

$\downarrow$ (9) $q_3\, 0\, D\, q_4$

$\downarrow$ (10) $q_4 \,|\, D\, q_4$

$\downarrow$ (11) $q_4 * D\, q_4$

(14) $q_6\, 0 \,|\, q_1 \longleftarrow$ (13) $q_5 \,|\, D\, q_6 \longleftarrow$ (12) $q_4\, 0 \,|\, q_5 \longleftarrow$ (11) $q_4 * D\, q_4$

(2)

(5)

Exercice 3 :

1)

a) Type inféré par CAML :

f : a' → int → 'a list → 'a list = <fun> (1)

b) f 5 3 [3;3;8;9;1] = 3 :: f 5 2 [3;8;9;1]

= 3 :: 3 :: f 5 1 [8;9;1] (1,5)

= 3 :: 3 :: 5 :: [8;9;1] = [3;3;5;8;9;1]

c) f 9 4 [7;8;2] = 7 :: f 9 3 [8;2]

= 7 :: 8 :: f 9 2 [2] (1,5)

= 7 :: 8 :: 2 :: f 9 1 []

= 7 :: 8 :: 2 :: 9 :: [] = [7;8;2;9]

d) f insere x dans z à la position y. (1)

2)
```
let rec coefficient_binomial i j =
      if j > i then 0
      else if j = 0 then 1
           else coefficient_binomial (i-1) (j-1)
                + coefficient_binomial (i-1) j ;;
```
(2)

coefficient_binomial : int → int → int = <fun>