USTHB – FEI - Département d'Informatique
LMD Licence 1ère Année MI 2016/2017
Module ''Programmation Fonctionnelle'' Section 6
Date : 07/05/2017

Corrigé de l'interrogation

Exercice 1 <7 points> :

1) Montrez que la fonction Reste3=λx.Reste3(x) est primitive récursive, Reste3(x) étant le reste de la division entière de x par 3
2) En déduire que l'ensemble Mult3 des entiers naturels multiples de 3 est primitif récursif
3) Montrer que la fonction f suivante est primitive récursive :

$$f=\lambda xy.\begin{cases} 1 \; si\; x\; est\; multiple\; de\; 3\; et\; x > y, \\ 0 \quad sinon \end{cases}$$

Les fonctions dont nous avons montré en cours qu'elles étaient PR, peuvent être considérées comme telles.

Solution :

1) On utilise la règle de récursion pour montrer que la fonction Reste3 est PR : Reste3 étant d'arité 1, il faut trouver deux fonctions PR g et h d'arités, respectivement, 0 et 2 de telle sorte que Reste3(0)=g et, pour tout entier naturel y, Reste3(y+1)=h(Reste3(y),y) :

   Reste3(0)=0 (fonction constante 0 d'arité 0)

   $$Rest3(y+1) = \begin{cases} 0\; si\; Reste3(y) = 2, \\ 1 + Reste3(y)\; sinon \end{cases}$$

   $$= \begin{cases} 0\; si\; moins\big(2, Reste3(y)\big) = 0, \\ 1 \quad sinon \end{cases} * \;(1+Reste3(y))$$

   =Sg(moins(2,Reste3(y))*(1+Reste3(y))

   $=[\otimes°(Sg°moins°(S°S°Z°P_2^2, P_1^2),\oplus(S°Z°P_2^2, P_1^2))](Reste3(y),y)$

   Conclusion :

   La fonction Reste3 est PR grâce à la dérivation PR suivante :

   f_1=0 PR car fonction constante d'arité 0
   $f_2=P_1^2$ PR car de base
   $f_3=P_2^2$ PR car de base
   f_4=Z PR car de base
   f_5=S PR car de base
   f_6=Sg PR (cours)
   f_7=moins PR (cours)
   $f_8=\oplus$ PR (cours)
   $f_9=\otimes$ PR (cours)
   $f_{10}=f_4°f_3$ PR (règle de composition appliquée aux fonctions h=f_4 et g=f_3)
   $f_{11}=f_5°f_{10}$ PR (règle de composition appliquée aux fonctions h=f_5 et g=f_{10})
   $f_{12}=f_5°f_{11}$ PR (règle de composition appliquée aux fonctions h=f_5 et g=f_{11})
   $f_{13}=f_7°(f_{12},f_2)$ PR (règle de composition appliquée aux fonctions h=f_7, $g_1=f_{12}$ et $g_2=f_2$)
   $f_{14}=f_6°f_{13}$ PR (règle de composition appliquée aux fonctions h=f_6 et g=f_{13})
   $f_{15}=f_8°(f_{11},f_2)$ PR (règle de composition appliquée aux fonctions h=f_8, $g_1=f_{11}$ et $g_2=f_2$)
   $f_{16}=f_9°(f_{14},f_{15})$ PR (règle de composition appliquée aux fonctions h=f_9, $g_1=f_{14}$ et $g_2=f_{15}$)
   f_{17}=Reste3 PR (règle de récursion appliquée aux fonctions g=f_1 d'arité 0 et h=f_{16} d'arité 2)

2) La fonction caractéristique de Mult3 est comme suit : pour tout entier naturel x,

   $$Car_{Mult3}(x) = \begin{cases} 1\; si\; x\; multiple\; de\; 3, \\ 0 \quad sinon \end{cases} = \begin{cases} 1\; si\; Reste3(x) = 0, \\ 0 \quad sinon \end{cases}$$

   $=\overline{sg}$ (Reste3(x)) $= (\overline{sg}°Reste3)(x)$.

Mr Isli (aisli@usthb.dz)

Conclusion :

La fonction caractéristique Car_{Mult3} est PR (règle de composition appliquée aux fonctions PR h= $\overline{sg}$ et g=Reste3). Mult3 est donc primitif récursif.

3) f(x,y) $=\begin{cases} 1 \ si \ x \ est \ multiple \ de \ 3 \ et \ x > y, \\ 0 \quad\quad\quad\quad\quad\quad\quad\quad\quad\quad sinon \end{cases}$

$=\begin{cases} 1 \ si \ x \ multiple \ de \ 3, \\ 0 \quad\quad\quad\quad\quad sinon \end{cases} * \begin{cases} 1 \ si \ x > y, \\ 0 \quad sinon \end{cases}$

$=Car_{Mult3}(x)*\begin{cases} 0 \ si \ moins(x,y) = 0, \\ 1 \quad\quad\quad\quad\quad sinon \end{cases}$

$= Car_{Mult3}(x)*Sg(moins(x,y))$

$=[\otimes\circ(Car_{Mult3}\circ P_1^2, Sg\circ moins)](x,y)$

Conclusion :

La fonction f est PR grâce à la dérivation PR suivante :

$f_1=P_1^2$ PR car de base

f_2=Sg PR (cours)

f_3=moins PR (cours)

$f_4=\otimes$ PR (cours)

$f_5= Car_{Mult3}$ PR (question 2)

$f_6=f_2\circ f_3$ PR (règle de composition appliquée aux fonctions $h=f_2$ et $g=f_3$)

$f_7=f_5\circ f_1$ PR (règle de composition appliquée aux fonctions $h=f_5$ et $g=f_1$)

$f_8=f_4\circ(f_7,f_6)=f$ PR (règle de composition appliquée aux fonctions $h=f_4$, $g_1=f_7$ et $g_2=f_6$)

Exercice 2 <7 points> :

Soit la machine de Turing MT=<S,E,I> définie comme suit :

- S={0,1}
- E={ q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_f }, q_0 et q_f étant, respectivement, l'état interne initial et l'état interne final de MT
- I={ (1) : q_010q_0, (2) : q_00Dq_1, (3) : q_101q_f, (4) : q_110q_2, (5) : q_20Dq_3, (6) : q_301q_4, (7) : q_41Gq_1, (8) : q_310q_6, (9) : q_60Dq_7, (10) : q_701q_8, (11) : q_81Gq_3, (12) : q_711q_0 }

1) Calculez MT(0), MT(1) et MT(2). En d'autres termes, déroulez la machine de Turing MT sur chacune des configurations initiales suivantes : $q_0\underline{0}$, $q_0\underline{1}$ et $q_0\underline{2}$
2) Quelle est la fonction calculée par la machine de Turing MT ?

Solution :

1) Déroulement de la machine de Turing MT :

$q_0\underline{0}=q_01\xrightarrow{(1)}q_00\xrightarrow{(2)}q_10\xrightarrow{(3)}q_f1=q_f\underline{0}$

$q_0\underline{1}=q_011\xrightarrow{(1)}q_001\xrightarrow{(2)}q_11\xrightarrow{(4)}q_20\xrightarrow{(5)}q_30\xrightarrow{(6)}q_41\xrightarrow{(7)}q_101\xrightarrow{(3)}q_f11=q_f\underline{1}$

$q_0\underline{2}=q_0111\xrightarrow{(1)}q_0011\xrightarrow{(2)}q_111\xrightarrow{(4)}q_201\xrightarrow{(5)}q_31\xrightarrow{(8)}q_60\xrightarrow{(9)}q_70\xrightarrow{(10)}q_81\xrightarrow{(11)}q_301\xrightarrow{(6)}q_411\xrightarrow{(7)}q_1011\xrightarrow{(3)}q_f111=q_f\underline{2}$

2) Pour x≥3, on pose n=x-3 :

$\underline{x}$=(x+1) barres=(4+(x-3)) barres=(4+n) barres=1^41^n

$q_0\underline{x}=q_01^41^n\xrightarrow{(1)}q_001^31^n\xrightarrow{(2)}q_11^31^n\xrightarrow{(4)}q_201^21^n\xrightarrow{(5)}q_31^21^n\xrightarrow{(8)}q_6011^n\xrightarrow{(9)}q_711^n\xrightarrow{(10)}q_011^n=q_01^{n+1}=q_0\underline{n}=q_0\underline{(x-3)}$

Conclusion :

La fonction f calculée par la machine de Turing vérifie donc :

$f(x)=\begin{cases} x \ si \ x \in \{0,1,2\}, \\ f(x-3) sinon \end{cases}$

La fonction f n'est rien d'autre que le reste de la division entière par 3, f=λx.(x mod 3)

Mr Isli (aisli@usthb.dz)

Exercice 3 <6 points> :

On considère les deux instructions CAML suivantes:

```
let rec mystere x=if x=[] then 0 else 2+mystere (tl x);;

let rec f x=if x=[] then [] else mystere (hd x)::f (tl x);;
```

1) Donnez pour chacune des fonctions le type inféré par CAML.
2) Déroulez la fonction mystere sur un exemple. Que fait mystere ?
3) Déroulez la fonction f sur un exemple. Que fait f ?

Solution :

1) mystere : ‘a list -> int = <fun>
   f : ‘a list list -> int list = <fun>
2) mystere [1 ;2 ;3] =2+mystere [2 ;3]
   =2+2+mystere [3]=2+2+2+mystere []=2+2+2+0=6
   La fonction mystere calcule le double du nombre d’éléments de la liste argument.
3) f [[1 ;2 ;3] ;[7 ;10] ;[1 ;3 ;9 ;0 ;2 ;8 ;9] ;[]]
   = mystere [1 ;2 ;3] :: f [[7 ;10] ;[1 ;3 ;9 ;0 ;2 ;8 ;9] ;[]]
   = mystere [1 ;2 ;3] ::mystere [7 ;10] :: f [[1 ;3 ;9 ;0 ;2 ;8 ;9] ;[]]
   = mystere [1 ;2 ;3] ::mystere [7 ;10] :: mystere [1 ;3 ;9 ;0 ;2 ;8 ;9] ::f [[]]
   = mystere [1;2;3] ::mystere [7;10] :: mystere [1;3;9;0;2;8;9] ::mystere [] ::f []
   = 6 ::4 ::14 ::0 :[]= [6 ; 4 ; 14 ; 0]

   La réponse que donnera CAML est donc comme suit :

   - : int list = [6; 4; 14; 0]

   La fonction f prend comme argument une liste de listes et distribue la fonction mystere sur les éléments de l’argument : elle remplace chaque liste élément de la liste argument, par le double de son nombre d’éléments.

Mr Isli (aisli@usthb.dz)