Université des Sciences et de la Technologie Houari Boumediène
Faculté d'Electronique et d'Informatique
Département d'Informatique
LMD Licence 1ère Année Mathématiques-Informatique 2012/2013
Module ''Programmation Fonctionnelle'' Section 1
Date : 14/05/2013

Corrigé de l'interrogation

Exercice 1 (1+4=5 points) :

1. Rappelez la règle de construction de fonctions primitives récursives dite 'règle de récursion'
2. Montrez que la fonction f=λxy.(x+y) ! est primitive récursive

Indication : les fonctions fact=λx.x!, plus=λxy.x+y et fois=λxy.x*y sont primitives récursives.

Solution :

1. Voir cours
2. f(x,0)=(x+0)!=fact(x)

   f(x,y+1) =(x+(y+1)) !=(x+y) !*(x+y+1)=fois(f(x,y),S(plus(x,y)))

   $=[fois°(P^3_2,S°plus°(P^3_1, P^3_3))](x,f(x,y),y)$

   La fonction f est donc primitive récursive. En voici une dérivation PR :

   f_1=fact PR

   f_2= P^3_2 PR car de base

   f_3= P^3_1 PR car de base

   f_4= P^3_3 PR car de base

   f_5=plus PR

   f_6=f5°(f3,f4) PR par application de la règle de composition

   f_7=S PR car de base

   f_8=f7°f6 PR par application de la règle de composition

   f_9=fois PR

   f_{10}=f9°(f2,f8) PR par application de la règle de composition

   f_{11}=f PR par application de la règle de récursion aux fonctions f_1 et f_{10}

Exercice 2 (5 points) :

Le but de l'exercice est de concevoir une machine de Türing calculant la fonction suivante :

$$f=\lambda x.\begin{cases}1\ si\ x\ pair,\\ \frac{x+1}{2}\ sinon\end{cases}$$

Il vous est demandé de procéder comme suit :

Etape 1 : répéter les deux points suivants jusqu'à ce qu'il n'y ait plus de barres

1. transformer la barre la plus à gauche en #
2. transformer la barre la plus à droite en un blanc (0)

Etape 2 : à l'issue de l'étape1, la machine doit être capable, selon que la dernière barre a été transformée en un # ou en un blanc, de déduire si l'entier de départ était pair ou impair. Cette information doit lui permettre de procéder à la deuxième étape qui consiste à transformer le contenu de son ruban de telle sorte qu'il contienne la représentation de l'image f(x) de l'entier x de départ.

Solution :

MT=<S,E,I>

- S={0,1,#}
- E={q_0,q_1,…,q_{19},q_f} avec q_0 état interne initial et q_f état interne final
- I={(1) $q_0$1#q_1, (2) q_1#Dq_2, (3) $q_2$1Dq_2, (4) $q_2$0Gq_3, (5) q_3##q_4,

  (6) q_4#0q_5, (7) $q_5$0Gq_4, (8) $q_4$01q_6, (9) $q_6$1Gq_7, (10) $q_7$01q_f,

  (11) $q_3$10q_8, (12) $q_8$0Gq_9, (13) q_9##q_{10}, (14) $q_9$1Gq_{11},(15) q_{11}1Gq_{11},

  (16) q_{11}#Dq_0,

  (17) q_{10}#1q_{12}, (18) q_{12}1Gq_{10}, (19) q_{10}01q_f}

<u>Discussion :</u>

- Si la machine arrive durant son calcul à une configuration avec q_3 comme état interne et la tête de lecture/écriture sur un #, le nombre de barres initiales est impair et l'entier de départ était pair. La machine passe alors dans l'état interne q_4 qui lui permet de procéder à l'exécution de la suite (6)-(7)-(8)-(9)-(10) d'instructions transformant le contenu du ruban en deux barres (représentation de 1) et faisant passer la machine dans l'état interne final q_f
- Si la machine arrive durant son calcul à une configuration avec q_9 comme état interne et la tête de lecture/écriture sur un #, le nombre de barres initiales est pair et l'entier de départ était impair. La machine passe alors dans l'état interne q_{10} qui lui permet de procéder à l'exécution de la suite (17)-(18)-(19) d'instructions transformant tous les # en barres avant de rajouter une barre et passer dans l'état final q_f
- Si la machine arrive durant son calcul à une configuration avec q_{11} comme état interne et la tête de lecture/écriture sur un #, elle a transformé autant de barres à gauche en # que de barres à droite en blanc. La machine exécute l'instruction (16) qui lui permet d'avancer d'une position à droite pour se placer sur la barre la plus à gauche et passer dans l'état interne initial q_0, réinitialisant de la sorte le processus de remplacement de la barre la plus à gauche par un # et de la barre la plus à droite par un blanc

Exercice 3 (0,5+0,5+0,25+0,5+0,5+0,5+0,5+0,25=3,5 points) :

Donnez pour chacune des instructions suivantes le résultat de son interprétation par CAML :

int_of_float 10.7+7;;

- : int = 17

sqrt 25+.3.5;;

Erreur (Cette expression est de type int, mais est utilisée avec le type float.)

"Alger".[0] ;;

- : char = `A`

(sub_string ''Alger'' 2 3)^(sub_string ''Tamenrasset'' 0 3);;

- : string = "gerTam"

let a="Alger";;

a : string = "Alger"

let a="Tamenrasset" and b=a in b^a;;

- : string = "AlgerTamenrasset"

"Alger".[1]::([]@[`m`;`8`]);;

- : char list = [`l`; `m`; `8`]

a;;

- : string = "Alger"

Exercice 4 (1,5 point) : Ecrire une fonction CAML n_div_p calculant le nombre de diviseurs pairs d'un entier.

Solution :

```
let rec n_div_p_se x i=     if i>x then 0
                            else     if x mod i=0 then 1+n_div_p_se x (i+2)
                                     else n_div_p_se x (i+2) ;;
let rec n_div_p x=if x=0 then -1 else n_d_v_p_se x 2;;
```

Le nombre de diviseurs pairs de 0 est infini : n-div_p 0 retourne -1.