

Examen Final

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

- 1) Montrer par récursion que la fonction $f_1 = \lambda xyz. x + y + z$ d'arité 3 est Primitive Récursive.
- 2) Montrer par composition que la fonction $f_2 = \lambda xyz. (x + y) * z$ d'arité 3 est Primitive Récursive.
- 3) Montrer par récursion que la fonction $f_3 = \lambda x. \sum_{k=0}^x k$ d'arité 1 est Primitive Récursive.
- 4) En déduire, par récursion, que la fonction $f_4 = \lambda xy. \sum_{k=0}^{\text{Max}(x,y)} k$ d'arité 2 est Primitive Récursive.

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

Exercice 2 (6,5 points = (4*0,75+0,5) +(1+2)) (Machines de Turing)

- 1) Soit la Machine de Turing $MT = \langle S, E, I \rangle$ avec $S = \{0, 1\}$, $E = \{q_0, q_1, q_2, q_3, q_4, q_f\}$ et $I = \{ (1) q_0 1 0 q_1, (2) q_1 0 D q_2, (3) q_2 0 1 q_f, (4) q_2 1 0 q_3, (5) q_3 0 D q_4, (6) q_4 1 1 q_f, (7) q_4 0 1 q_f \}$
 - a) Dérouler la machine pour : i) 0 ii) 1 iii) 3 iv) 4.
 - b) Que fait la machine ?
- 2) On veut concevoir une machine de Turing calculant la fonction $f = \lambda x. \text{Sg}(x)+1 = \lambda x. \begin{cases} 1 & \text{si } x = 0, \\ 2 & \text{sinon} \end{cases}$
 - a) Expliquer la démarche à suivre.
 - b) Donner l'ensemble I des instructions de la machine.

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

- 1) On considère les fonctions CAML suivantes :

```
let rec f x = if x < 0 then 0 else x + f (x-1) ;;
let rec g y = if y = [] then [] else f (hd y) :: g (tl y) ;;
```

 - a) Donner les types de f et g inférés par CAML.
 - b) Dérouler la fonction g pour : i) $y = [1 ; 2]$ ii) $y = [3 ; -2 ; 4]$
 - c) Que fait la fonction g ?
- 2) Ecrire une fonction *Clean* en CAML qui permet de supprimer toutes les occurrences de la chaîne vide d'une liste de chaînes de caractères.

Exemples :

```
Clean ["USTHB" ; "Alger" ; "" ; "Turing"] = ["USTHB" ; "Alger" ; "Turing"]
Clean ["USTHB" ; "" ; "FI" ; ""] = ["USTHB" ; "FI"]
```

...Bon courage...

Final Exam

Exercise 1 (6,5 points = 1,5+1,5+1,5+2) (Recursive Functions)

- 1) Show by recursion that the function $f_1 = \lambda xyz. x + y + z$ of arity 3 is Primitive Recursive.
- 2) Show by composition that the function $f_2 = \lambda xyz. (x + y) * z$ of arity 3 is Primitive Recursive.
- 3) Show by recursion that the function $f_3 = \lambda x. \sum_{k=0}^x k$ of arity 1 is Primitive Recursive.
- 4) Deduce, by recursion, that the function $f_4 = \lambda xy. \sum_{k=0}^{Max(x,y)} k$ of arity 2 is Primitive Recursive.

Indication: the following functions are PR : \oplus , \otimes , Max, Min, Moins, Fact, Abs, Exp, Sg, \overline{Sg} , Pred, mod, div, the constant functions of arity 0, the constant functions of arity 1.

Exercise 2 (6,5 points = (4*0,75+0,5) +(1+2)) (Turing Machines)

- 1) We consider a Turing machine $MT = \langle S, E, I \rangle$ with $S = \{0, 1\}$, $E = \{q_0, q_1, q_2, q_3, q_4, q_f\}$ and $I = \{ (1) q_0 10 q_1, (2) q_1 0D q_2, (3) q_2 01 q_f, (4) q_2 10 q_3, (5) q_3 0D q_4, (6) q_4 11 q_f, (7) q_4 01 q_f \}$
 - a) Trace the machine for: i) 0 ii) 1 iii) 3 iv) 4.
 - b) What does the machine compute?
- 2) We want to conceive a Turing machine computing the function $f = \lambda x. Sg(x)+1 = \lambda x. \begin{cases} 1 & \text{if } x = 0, \\ 2 & \text{otherwise} \end{cases}$
 - a) Explain your method.
 - b) Write the set I of the machine's instructions.

Exercise 3 (7 points = ((2*1+2*1+1)+2) (CAML Language)

- 1) We consider the following CAML functions:


```
let rec f x = if x < 0 then 0 else x + f (x - 1) ;;
let rec g y = if y = [] then [] else f (hd y) :: g (tl y) ;;
```

 - a) Write the types of f and g as inferred by CAML.
 - b) Trace the function g for: i) $y = [1 ; 2]$ ii) $y = [3 ; -2 ; 4]$
 - c) What does the function g do?
- 2) Write a function *Clean* in CAML that removes all occurrences of the empty string from a list of strings.

Examples:

```
Clean ["USTHB" ; "Alger" ; "" ; "Turing"] = ["USTHB" ; "Alger" ; "Turing"]
Clean ["USTHB" ; "" ; "FI" ; ""] = ["USTHB" ; "FI"]
```

...*Good luck*...