Examen de Programmation Fonctionnelle
(3 exercices, 2 pages)

Répondre sur des feuilles volantes, comme avec un examen classique ; numéroter toutes les pages ; scanner (imprimante-scanner ou téléphone) ou prendre en photo (téléphone) toutes les pages ; m'envoyer le scan (sous forme de fichiers pdf) ou les photos (fichiers jpg) à a_isli@yahoo.com ; la note du contrôle continu (interrogation) sera extraite des deux meilleures réponses aux trois exercices. Tout non-respect du délai de 24 heures sera sanctionné : le travail doit me parvenir demain jeudi 15/10/2020, au plus tard à 18h30.

Exercice 1 :

1) Montrer que la fonction f suivante est primitive récursive :

$$f = \lambda xy. \begin{cases} x + y & \text{si } x \geq y, \\ y & \text{sinon} \end{cases}$$

Donner une dérivation PR de f .

2) Montrer que l'ensemble $I = \{(x,y) \in \mathbb{N}^2 : x \geq y \text{ et } x+y \text{ pair}\}$ est primitif récursif.

Indication : Les fonctions vues en cours peuvent être considérées primitives récursives (exp, Sg, $\overline{Sg}$, moins, Abs, min, max, fact, S, Pred, $\oplus$, $\otimes$, $q_2 = \lambda x.x \text{ div } 2$, $r_2 = \lambda x.x \text{ mod } 2$, div, mod, $C_k = \lambda.k$, $F_k = \lambda x.k$, avec k constante)

Exercice 2 :

1) Concevoir une machine de Turing calculant toutes les fonctions suivantes :

- a) la fonction $P_1^1 = \lambda x.x$
- b) la fonction $\oplus = \lambda xy.(x+y)$
- c) les fonctions $f_p = \lambda x_1 \dots x_p.(0, x_1, \dots, x_p)$, avec p pair et $p \geq 4$
- d) les fonctions $f_p = \lambda x_1 \dots x_p.1$, avec p impair et $p \geq 3$

2) Soit la machine de Turing $MT = \langle S, E, I \rangle$ définie par :

- $S = \{0, 1, *\}$
- $E = \{q_0, q_1, q_2, q_3, q_4, q_5, q_f\}$
- $I = \{(1) q_0 1 D q_0, (2) q_0 * D q_0, (3) q_0 0 G q_1, (4) q_1 1 0 q_1, (5) q_1 0 G q_2, (6) q_2 1 1 q_1, (7) q_2 * D q_3, (8) q_2 0 1 q_f, (9) q_3 0 1 q_4, (10) q_4 1 G q_4, (11) q_4 * G q_4, (12) q_4 0 D q_f\}$

- a) Dérouler la machine sur les entrées suivantes : 2, (1,2) et (2,0,1).
- b) Quelles sont les fonctions calculables par la machine.

3) Soient les machines de Turing MT_{succ} et $MT_{\text{mod}2}$ suivantes :

$$MT_{\text{succ}} = \langle S_1, E_1, I_1 \rangle :$$

- $S_1 = \{0, 1\}$
- $E_1 = \{q_0, q_1, q_f\}$
- $I_1 = \{(1) q_0 1 G q_1, (2) q_1 0 1 q_f\}$

$$MT_{\text{mod}2} = \langle S_2, E_2, I_2 \rangle :$$

- $S_2 = \{0, 1\}$
- $E_2 = \{q_0, q_1, q_2, q_3, q_4, q_5, q_f\}$
- $I_2 = \{(1) q_0 1 0 q_1, (2) q_1 0 D q_2, (3) q_2 0 1 q_f, (4) q_2 1 0 q_3, (5) q_3 0 D q_4, (6) q_4 0 1 q_5, (7) q_5 1 G q_2, (8) q_4 1 1 q_0\}$

Le but est d'illustrer la composition de machines de Turing à travers MT_{Succ} et MT_{mod2} :

- a) Donner la machine $MT = \langle S, E, I \rangle$ égale à la composition $MT_{\text{mod2}} \circ MT_{\text{Succ}}$ des machines MT_{Succ} et MT_{mod2} .
- b) Expliquer comment sont obtenus les ensembles S, E et I. Illustrer à travers le déroulement de MT sur l'entrée 3.

Exercice 3 :

Si les deux fonctions demandées utilisent d'autres fonctions, il faut définir ces dernières.

- 1) Ecrire une fonction CAML **sc x y** testant si une chaîne de caractères x est sous-chaîne d'une chaîne de caractères y.

Exemples : sc "ab" "ab"=vrai ; sc "ab" "abc"=vrai ; sc "ab" "cab"=vrai ; sc "ab" "xyabz"=vrai ; sc "" x=vrai (x chaîne de caractères quelconque) ; sc "ab" "xacby"=faux

- 2) Ecrire une fonction CAML **car_nocc x** extrayant, à partir d'une chaîne de caractères x, une liste $[(c_1, n_1), \dots, (c_m, n_m)]$, $c_1, \dots, c_m$ étant (tous) les caractères occurrant dans x, et $n_1, \dots, n_m$ leurs nombres (non nuls) respectifs d'occurrences.

Exemples : car_nocc ""=[] ; car_nocc "usthb"=[(u,1);(s,1);(t,1);(h,1);(b,1)];
car_nocc "2020"=[(2,2);(0,2)]