Licence STS, semestre 3 — 2007–2008
Algorithmique et approche fonctionnelle (Info 211) — Durée : 2h
2 pages

Examen du 5 Novembre 2007

Les notes de cours, de TD et de TP manuscrites ainsi que les supports de cours distribués cette année sont les seuls documents autorisés.

Veuillez lire attentivement les questions. Veuillez rédiger proprement, clairement et de manière concise et rigoureuse. Le barème est indicatif.

1 Typage (4 points)

Les fonctions suivantes sont-elles bien typées ? Si oui, donner leur type, sinon préciser pourquoi.

```
let f1 x = not x

let f2 x = [(x + 1); 2.0]

let f3 x y = if y then x

let f4 x y = x

let f5 p a b = b <= 0 || p(a)

let rec f6 x = if true then x::[] else x@(f6 x)
```

2 Évaluation (4 points)

1. Étant donnés le type `t` et la fonction `mystere: t list -> int * int` suivants :

   ```
   type t = A | B

   let rec mystere l =
     match l with
         [] -> (0,0)
       | A::s -> let a,b = mystere s in (a+1,b)
       | B::s -> let a,b = mystere s in (a,b+1)
   ```

   Donner le résultat de l'évaluation de `mystere [A;A;B;B;B;A;B;B]`
2. Étant donnée la fonction `mystere: 'a list -> 'a list` suivante :

   ```
   let rec mystere l =
     match l with
         [] -> []
       | [x] -> [x]
       | x::y::s -> y::(mystere (x::s))
   ```

Donner le résultat de l'évaluation de `mystere [1;2;3;4]`

3. Étant donnée la fonction `mystere: int -> int -> int` suivante :

```
let rec mystere acc n =
  match n with
      0 -> acc
    | x -> mystere (x+acc) (x-1)
```

Donner le résultat de l'évaluation de `mystere 0 5`

3 Problème : Arithmétique de Peano (12 points)

En 1889, Guiseppe Peano proposa une axiomatisation des entiers naturels décrite informellement par les cinq axiomes suivants :

- l'élément appelé zéro et noté : 0, est un entier naturel ;
- tout entier naturel n a un unique successeur, noté $S(n)$;
- aucun entier naturel n'a 0 pour successeur ;
- deux entiers naturels ayant même successeur sont égaux ;
- si un ensemble d'entiers naturels contient 0 et contient le successeur de chacun de ses éléments, alors cet ensemble est égal à $\mathbb{N}$.

Dans la suite du problème, nous représenterons en OCaml les entiers naturels de Peano par des éléments de type `unit list`. Le nombre n d'éléments `()` de la liste représentant le nombre de successeurs de 0. Par exemple, la valeur 0 est représentée par `[]`, la valeur 3 par `[();();()]`.

1. Définir la fonction `succ: unit list -> unit list` qui, étant donnée une liste `l` représentant l'entier de Peano n, retourne la liste qui représente $S(n)$.
2. Définir les fonctions de conversion entre les entiers OCaml et entiers de Peano :
   ```
   val int_of_peano: unit list -> int
   val peano_of_int: int -> unit list
   ```
   La fonction `peano_of_int` n'étant définie que pour des entiers positifs ou nuls, prévoir un cas d'erreur.
3. Donner une définition récursive terminale des fonctions précédentes.
4. Sachant que la valeur `max_int : int` représente le plus grand entier du langage OCaml, est-ce que `succ (peano_of_int max_int)` est égal à `peano_of_int (max_int + 1)` ?
5. Définir l'addition et la multiplication sur les entiers de Peano :
   ```
   val add: unit list -> unit list -> unit list
   val mult: unit list -> unit list -> unit list
   ```
6. Soient n_1, n_2 deux entiers de Peano, définir la relation d'ordre $n_1 \leq n_2$ par récursion sur n_1 et n_2 :
   ```
   val inf_eq: unit list -> unit list -> bool
   ```