

Université PIERRE ET MARIE CURIE

—O—

DEUG MIAS

Types et structures de données

Examen

28 Janvier 2004

Documents autorisés : tous supports papier.

Durée : 2 heures.

Remarques :

- il est parfois nécessaire pour définir une fonction d'avoir à définir d'autres fonctions ou valeurs auxiliaires ;*
- si besoin est, on peut utiliser une fonction ou valeur demandée à une question précédente sans avoir pour autant répondu cette question.*

EXERCICE I Une liste de couples d'entiers représente une partie de dominos si elle satisfait aux conditions suivantes :

1. On n'a joué que des dominos : i.e. tous les entiers présents dans la liste sont compris entre 0 et 6 (inclusivement).
2. On n'a pas triché : i.e. aucun domino n'apparaît deux fois dans la liste (dans un sens ou l'autre).
3. On a respecté les règles : i.e. si deux dominos (n_1, n_2) et (m_1, m_2) se suivent dans la liste alors $n_2 = m_1$.

QUESTION (I.1) Donnez la définition O'CAML d'un prédicat binaire

`is_dom : int -> int -> bool`

tel que `(is_dom n1 n2)` vaut `true` si le couple `(n1,n2)` représente un domino et `false` sinon.

QUESTION (I.2) Donnez la définition O'CAML d'un prédicat d'arité 4

`diff_dom : int -> int -> int -> int -> bool`

tel que `(diff_dom n1 n2 m1 m2)` vaut `true` si le domino `(n1,n2)` est différent du domino `(m1,m2)` et `false` sinon.

QUESTION (I.3) Donnez la définition O’CAML d’un prédicat ternaire

`mem_dom : int -> int -> (int * int) list -> bool`

tel que `(mem_dom n1 n2 ds)` vaut `true` si le domino `(n1,n2)` appartient à la liste `ds` et `false` sinon.

QUESTION (I.4) Donnez la définition O’CAML d’un prédicat sur les listes de couples d’entiers qui donne `true` si et seulement si son argument représente effectivement une partie de dominos.

QUESTION (I.5) Donnez le type de la fonction que vous avez définie.

QUESTION (I.6) Si une liste de couples d’entiers représente une partie de dominos, elle contient une information redondante : le même nombre est donné deux fois (condition 3 ci-dessus). On peut donc compacter une telle liste en supprimant cette redondance. Par exemples : `[(4,5)]` est compactée en `[4 ; 5]` ; `[(4,5) ; (5,0)]` est compactée en `[4 ; 5 ; 0]` et `[(4,5) ; (5,0) ; (0,1)]` est compactée en `[4 ; 5 ; 0 ; 1]`.

Donnez la définition O’CAML d’une fonction `compact` qui prend en argument une liste de couples d’entiers, supposée représenter une partie de dominos, et qui renvoie sa représentation compactée.

QUESTION (I.7) Donnez le type de la fonction `compact` que vous avez définie.

EXERCICE II Quelques manipulations de tableaux.

QUESTION (II.1) Donnez la définition O’CAML d’une fonction

`array_concat : 'a array array -> 'a array`

qui concatène les tableaux contenus dans un tableau. Par exemple

`(array_concat [| [|1 ; 2|] ; [|3|] ; [|] ; [|4 ; 5 ; 6|] |])`

vaudra `[|1 ; 2 ; 3 ; 4 ; 5 ; 6|]`.

QUESTION (II.2) Donnez la spécification d’une fonction `insert_list` qui prend en argument un tableau `t`, une liste `xs`, un entier `i` et qui insère les éléments de `xs` dans le tableau `t` à partir de la position `i`.

NB : veillez à ce que votre spécification n’omette aucun cas.

QUESTION (II.3) Donnez l’implantation de ce que vous avez spécifié.

EXERCICE III On considère un fichier texte contenant deux sortes de lignes :

1. Des lignes de *commentaires* dont le premier caractère est `#`.
2. Des lignes *régulières* dont le premier caractère n’est pas `#`.

Un paragraphe de commentaire est une suite de lignes de commentaire consécutives. Un paragraphe régulier est une suite de lignes régulières consécutives. Le but de la manœuvre est d’inverser dans un fichier donné les paragraphes de commentaire et les paragraphes

réguliers sans pour autant modifier l'ordre des lignes dans les paragraphes. Exemple : le

| | | | |
|-------------------|-----------------|-----------|-----------------|
| fichier contenant | # commentaire 1 | deviendra | texte 1 |
| | texte 1 | | texte 2 |
| | texte 2 | | # commentaire 1 |
| | # commentaire 2 | | texte 3 |
| | # commentaire 3 | | # commentaire 2 |
| | texte 3 | | # commentaire 3 |

QUESTION (III.1) Donnez la définition O'CAML d'une fonction

```
prefix : ('a -> bool) -> 'a list -> ('a list * 'a list)
```

telle que `(prefix p [x1; ..; xn; y1; ..; ym])` vaut `([x1; ..; xn], [y1; ..; ym])` avec `(p xi)` vaut `true` pour tout $i \in [1..n]$ et `(p y1)` vaut `false`.

QUESTION (III.2) Une ligne vide est-elle une ligne de commentaire ?

QUESTION (III.3) Donnez la définition O'CAML d'un prédicat sur les chaînes de caractères `is_comment` tel que `(is_comment ln)` renvoie `true` ssi `ln` est une ligne de commentaire.

QUESTION (III.4) Donnez les définitions O'CAML de deux fonctions `split_comments` et `split_regulars` telles que :

- `(split_comments xs)` vaut `(cs, ys)` avec : `xs = cs@ys`, `cs` ne contient que des lignes de commentaires et `ys` ne commence pas par une ligne de commentaire ;
- `(split_regulars xs)` vaut `(rs, ys)` avec : `xs = rs@ys`, `rs` ne contient que des lignes régulières et `ys` ne commence pas par une ligne régulière.

QUESTION (III.5) Donnez le type des deux fonctions que vous avez définies.

QUESTION (III.6) Donnez la définition O'CAML d'une fonction `invert` qui prend en argument une liste de chaînes de caractères et renvoie la liste où les paragraphes de commentaires et les paragraphes réguliers ont été inversés.

QUESTION (III.7) Donnez la définition O'CAML d'une fonction

```
input_file : string -> string list
```

telle que `(input_file fname)` renvoie la liste des lignes contenues dans le fichier nommé `fname`.

QUESTION (III.8) Donnez la définition O'CAML d'une fonction

```
output_file : string -> string list -> unit
```

telle que `(output_file fname lns)` crée un fichier nommé `fname` et y écrit les lignes `lns`.

QUESTION (III.9) Donnez la définition O'CAML d'une fonction

```
invert_file : string -> string -> unit
```

telle que `(invert_file fname1 fname2)` crée le fichier `fname2` contenant les paragraphes de `fname1` mais où les paragraphes de commentaires et les paragraphes réguliers ont été inversés.