

Contrôle terminal de Programmation Fonctionnelle

Durée : 2h. Documents autorisés : Notes personnelles de cours et de TD. Les fonctions sont à écrire en OCAML. Attention à ne pas utiliser d'éléments impurs ! Chaque question peut être traitée indépendamment des autres.

I Récursivité et filtrage

A On considère des vecteurs de grande dimension, à composantes réelles, et ayant la particularité de comporter beaucoup de composantes nulles. De tels vecteurs sont appelées *vecteurs creux*.

En OCAML, on choisit de représenter un vecteur creux $v = (v_1 \ v_2 \ \dots \ v_m)$ par un couple (m, lv) où m est la dimension du vecteur et lv est une liste de couples (j, v_j) où v_j est une composante non nulle du vecteur et j est le numéro de cette composante ($j \in \{1, \dots, m\}$). **La liste est ordonnée par ordre croissant des numéros de composante.**

Exemple : le vecteur $(0 \ 0 \ 4 \ 0 \ 0 \ 0 \ 6 \ 0 \ 0 \ 0)$ est représenté par :

$(10, [(3, 4.); (7, 6.)])$

Les fonctions à écrire doivent renvoyer une erreur (à l'aide de `failwith`) lorsque leurs arguments sont invalides.

1. De quel type sont les valeurs OCAML qui représentent un vecteur creux ?
2. Définir une fonction `composante` qui retourne la j -ième composante d'un vecteur creux.

Exemples :

`composante (10, [(3, 4.); (7, 6.)]) 2 = 0.`

`composante (10, [(3, 4.); (7, 6.)]) 3 = 4.`

3. Définir une fonction `add` qui additionne 2 vecteurs creux.

Rappel : $u + v = (u_1 + v_1 \ u_2 + v_2 \ \dots \ u_m + v_m)$.

Exemple :

`add (10, [(3, 4.); (7, 6.)]) (10, [(1, 1.); (3, 3.)]) =
 (10, [(1, 1.); (3, 7.); (7, 6.)])`

$(0 \ 0 \ 4 \ 0 \ 0 \ 0 \ 6 \ 0 \ 0 \ 0) + (1 \ 0 \ 3 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0) = (1 \ 0 \ 7 \ 0 \ 0 \ 0 \ 6 \ 0 \ 0 \ 0)$

B En s'inspirant de la représentation des vecteurs creux, on représente une *matrice creuse* i.e. comportant un grand nombre de coefficients nuls, en utilisant la représentation de ses vecteurs lignes : la matrice $A = (a_{ij})$ à n lignes et m colonnes ($i \in \{1, \dots, n\}, j \in \{1, \dots, m\}$) est représentée par le couple $((n, m), l)$ où l est une liste de couples (i, lv_i) avec i , un indice de ligne et lv_i , une liste de couples (j, a_{ij}) où j est un indice de colonne et a_{ij} un coefficient non nul. **Les listes sont ordonnées par ordre croissant des indices.**

Exemple : la matrice $\begin{pmatrix} 0 & 0 & 7 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 \end{pmatrix}$ est représentée par :

$((4, 6), [(1, [(3, 7.); (6, 4.)]); (4, [(3, 5.)])])$

1. De quel type sont les valeurs OCAML qui représentent une matrice creuse ?
2. Ecrire une fonction `transpose` qui retourne la transposée d'une matrice creuse. Rappel : la transposée de A est la matrice à m lignes et n colonnes telle que $A^t = (a_{ji})$. Exemple :

`transpose ((4, 6), [(1, [(3, 7.); (6, 4.)]); (4, [(3, 5.)])]) =`
`((6, 4), [(3, [(1, 7.); (4, 5.)]); (6, [(1, 4.)])])`

$$\begin{pmatrix} 0 & 0 & 7 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 \end{pmatrix}^t = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 7 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \end{pmatrix}$$

II Lambda-calcul

Réduire chacune des λ -expressions suivantes jusqu'à obtenir une λ -expression sous forme normale. (Le choix du redex n'est pas imposé.) Indiquer le redex que vous choisissez avec le symbole λ . Préciser la règle utilisée (α - ou β -conversion) et n'utiliser la α -conversion qu'en cas de problème de capture de variable.

1. $\lambda n. \lambda m. (m \ n) \ \lambda f. \lambda x. (f \ x) \ \lambda f. \lambda x. x$
2. $\lambda n. \lambda m. (m \ n) \ \lambda f. \lambda x. (f \ (f \ x)) \ \lambda f. \lambda x. (f \ x)$
3. $\lambda n. \lambda f. \lambda x. (f \ (n \ f \ x)) \ \lambda f. \lambda x. (f \ x) \ \lambda f. (f \ x)$