

## TD/TP 3a : programmation fonctionnelle, OCaml

---

### TRÈS IMPORTANT : vérification des programmes

Testez systématiquement tous les exemples OCaml qui ont été donnés. Testez, avec encore plus de soin, les petits programmes OCaml que vous avez vous-même écrit. Vous devez considérer comme faux un programme que vous n'avez pas testé.

### PRÉPARATION :

- Télécharger l'archive « `TPMobile.zip` ». Désarchiver ce fichier. Cette archive contient quatre fichiers qui ne doivent pas être renommés :
  - Un fichier « `Makefile` » qui est utilisé pour compiler votre code et générer le fichier exécutable nommé « `MobileExecutable` ». Ce fichier peut être consulté, mais pas modifié !
    - Pour compiler votre code et générer le fichier exécutable vous devez utiliser la commande : « `make` ».
      - Attention, pour cela vous devez utiliser le terminal et vous devez vous placer dans le répertoire « `TPMobile` » qui contient les trois fichiers téléchargés.
    - Pour exécuter votre code vous devez exécuter la commande « `./MobileExecutable` ».
    - Pour effacer les fichiers générés automatiquement, vous devez exécuter la commande « `make clean` ».
    - Si vous n'arrivez pas à exécuter la commande « `make` » dans votre terminal (par exemple, vous utilisez l'environnement Windows), vous devez exécuter la suite de commandes :
      - `ocamlc -c Mobile_sig.mli`
      - `ocamlc -c Mobile.ml`
      - `ocamlc -c TPMobile.ml`
      - `ocamlc -o MobileExecutable.exe Mobile.cmo TPMobile.cmo`
      - `./MobileExecutable.exe`à chaque fois que vous souhaitez compiler et exécutez votre programme.
  - Trois fichiers « `TPMobile.ml` », « `Mobile.ml` », « `Mobile_sig.ml` » que vous devrez compléter en répondant aux questions des exercices ci-dessous.
    - Attention, n'oubliez pas de supprimer les commentaires qui entourent une fonction ou un prédicat que vous avez développé pour pouvoir le/la tester.

- Pour répondre à certaines questions vous serez probablement amenés à développer des fonctions supplémentaires, n'hésitez pas à les créer.

## Partie 1 : Mobile (types récurrents)

Dans cet exercice, nous nous intéressons à la modélisation de mobiles décoratifs tels que celui de la figure suivante :



Ces mobiles sont constitués d'une barre suspendue à un fil, aux extrémités de laquelle sont suspendus soit un mobile, soit un élément décoratif. Tout le problème de la construction de tels mobiles est de s'assurer que les différentes barres restent horizontales (nous parlons d'équilibre du mobile).

**Remarque :** ces mobiles ont une structure « récursive ».

Comme nous l'avons vu, la propriété de base d'un mobile est l'équilibre statique. Si nous négligeons le poids des fils de suspension ainsi que celle des barres, un mobile est en équilibre si tous ses éléments respectent la loi des moments :

$$P_{M_g} \cdot L_g = P_{M_d} \cdot L_d \quad (*)$$

où

- $P_{M_g}$  (resp.  $P_{M_d}$ ) est le poids du mobile suspendu à l'extrémité gauche (resp. droite) et
- $L_g$  (resp.  $L_d$ ) la longueur entre le point de suspension et l'extrémité gauche (resp. droite).

Un mobile a une des deux formes suivantes :

- soit c'est un objet décoratif, sa modélisation peut se limiter à son poids.
- soit c'est une barre avec deux mobiles suspendus aux extrémités. Il sera modélisé par un quadruplet : (longueur, ratio, mobgauche, mobdroit), où
  - longueur est la longueur totale de la barre ( $L_g + L_d$ ),
  - ratio est le rapport qui vérifie  $L_g = \text{ratio} \times \text{longueur}$  et  $L_d = (1 - \text{ratio}) \times \text{longueur}$ .

Par convention, nous mettrons toujours le mobile le plus lourd à gauche, ce qui induira  $0 \leq \text{ratio} \leq 0.5$ ).

Le poids d'un mobile « barre » est égale à la somme des poids des mobiles suspendus (sous l'hypothèse que le poids de la barre est négligeable).

En OCaml, nous représenterons un mobile par une valeur du type :

```
type mobile =  
  | Point of float  
  | Barre of float * float * mobile * mobile;;  
      (* longueur, ratio, mobg,   mobd *)
```

## Exercice 1 (un premier mobile)

### Question 1 :

Déclarez type `mobile` dans le module `Mobile_sig` et définissez-le dans le module `Mobile`.

### Question 2 :

Exprimez le ratio d'un mobile équilibré en fonction des poids de ses mobiles gauche et droit. Explicitez votre raisonnement mathématique dans les commentaires au début du fichier `TPMobile.ml`. Notations à utiliser dans vos formules :

- `ratio`, `longueur` ;
- `Lg` - longueur entre le point de suspension et l'extrémité gauche ;
- `Ld` - longueur entre le point de suspension et l'extrémité droite ;
- `PMg` - poids du mobile suspendu à l'extrémité gauche ;
- `PMd` - poids du mobile suspendu à l'extrémité droite.

Déclarez une fonction « `ratio1` » de type : « `float -> float -> float` » dans module `Mobile_sig`.

Définissez cette fonction dans le module `Mobile`. La fonction « `ratio1` » calcule le ratio d'un mobile équilibré en fonction des poids de ses mobiles gauche et droit en utilisant la formule que vous avez précédemment obtenue.

### Question 3 :

Dans les commentaires du fichier `TPMobile.ml` dessinez le mobile `m1` ci-dessous (« `u1` » représente une unité de longueur) :

- une barre de 200 `u1` à laquelle sont suspendus
  - (à droite) un objet de 300 `g`
  - (à gauche) une barre de 150 `u1` à laquelle sont suspendus
    - (à droite) un objet de 250 `g`
    - (à gauche) une barre 100 `u1` à laquelle sont suspendus
      - (à droite) un objet de 200 `g` et
      - (à gauche) un objet de 300 `g`.

Vous trouverez un exemple de dessin d'un mobile dans le fichier `TPMobile.ml`.

#### Question 4 :

À l'aide de la fonction « `ratio1` » précédemment définie, calculez le ratio pour chaque barre du mobile `m1`. À noter que ce mobile contient trois barres, il faudra donc créer trois variables `r1`, `r2` et `r3` dans le fichier `TPMobile.ml`.

### Exercice 2 (construction du mobile)

L'objet des fonctions suivantes est de construire des mobiles qui sont équilibrés.

#### Question 1 :

Déclarez une fonction `point` de type `float -> mobile` dans le module `Mobile_sig`. Définissez cette fonction dans le module `Mobile`. La fonction `point`, étant donné un poids, construit un mobile élémentaire.

#### Question 2 :

Déclarez une fonction `barre` de type `float -> float -> mobile -> mobile -> mobile` dans le module `Mobile_sig`.

Définissez cette fonction dans le module `Mobile`. La fonction `barre`, étant donné une longueur, un ratio, un mobile gauche et un mobile droit, construit un mobile qui représente une barre à laquelle les mobiles gauche et droit sont suspendus.

#### Question 3 :

En utilisant les fonctions `point` et `barre` ainsi que les variables `r1`, `r2` et `r3` définies précédemment, créez le mobile `m1` ci-dessous (à noter que vous avez déjà étudié ce mobile dans la question 3 de l'exercice 1) dans le fichier `TPMobile.ml`:

- une barre de 200 `ul` à laquelle sont suspendus
  - (à droite) un objet de 300 `g`
  - (à gauche) une barre de 150 `ul` à laquelle sont suspendus
    - (à droite) un objet de 250 `g`
    - (à gauche) une barre 100 `ul` à laquelle sont suspendus
      - (à droite) un objet de 200 `g` et
      - (à gauche) un objet de 300 `g`.

À noter que vous pouvez afficher un mobile en utilisant la fonction `print_mobile` définie dans le module `Mobile`.

#### Question 4 :

Déclarez une fonction `poids` de type `mobile -> float` dans le module `Mobile_sig`. Définissez cette fonction dans le module `Mobile`. La fonction `poids`, étant donné un mobile, calcule son poids total.

#### Question 5 :

Déclarez une fonction `ratio2` de type `mobile -> mobile -> float` dans le module `Mobile_sig`.

Définissez cette fonction dans le module `Mobile`. La fonction `ratio2`, étant donné deux mobiles, calcule le rapport qui permet de déterminer les longueurs  $L_g$  et  $L_d$  garantissant l'équilibre. Vous pouvez utiliser la fonction `ratio1` définie précédemment.

#### Question 6 :

Déclarez une fonction `construit_mobile` de type `float -> mobile -> mobile -> mobile` dans le module `Mobile_sig`.

Définissez cette fonction dans le module `Mobile`. La fonction `construit_mobile`, étant donné la longueur de la barre et deux mobiles, construit le mobile équilibré correspondant (vous prendrez soin de mettre le mobile le plus lourd à gauche).

#### Question 7 :

Dans le fichier `TPMobile.ml` définissez le mobile `m1_bis` en redéfinissant le mobile `m1` de la question 3 et en utilisant les fonctions `construit_mobile` et `point`. Vérifiez que vous obtenez un résultat identique.

### Exercice 3 (vérification de l'équilibre d'un mobile)

#### Question 1 :

Déclarez un prédicat `equilibre` de type `mobile -> bool` dans le module `Mobile_sig`.

Définissez ce prédicat dans le module `Mobile`. Le prédicat `equilibre` teste si un mobile donné est équilibré.

**Rappel :** Un mobile est équilibré si son ratio vérifie l'équation d'équilibre (\*).

#### Question 2 :

Dans le fichier `TPMobile.ml` définissez le mobile `m2` ci-dessous en utilisant les fonctions `construit_mobile` et `point` :

- une barre de 100 `ul` à laquelle sont suspendus
  - (à droite) un objet de 100 `g` et
  - (à gauche) un objet de 200 `g`.

Ce mobile doit être équilibré. Votre prédicat donne-t-il la bonne réponse ? Sinon, pourquoi, et dans ce cas, pouvez-vous proposer une solution corrective en créant une fonction `equilibre_bis` qui est la version corrigée de la fonction `equilibre` ?