

TP de *Logique et Circuits* n° 2**Introduction à Caml (2)**

Exercice 1 Définir un type `couple` (avec constructeur) dont les éléments sont des couples de type arbitraire. Définir une fonction qui teste l'égalité des deux éléments d'un couple. Cette fonction peut-elle s'appliquer à n'importe quel couple ?

Exercice 2 Définir le type `rationnel` en Caml (couple numérateur / dénominateur). Écrire une fonction construisant un rationnel à partir d'un couple de flottants (x, y) avec y non nul, et déclenchant une erreur lorsque y est nul. Écrire une fonction qui teste l'égalité de deux rationnels (effectuer le produit croisé entre numérateurs et dénominateurs). Écrire des fonctions qui réalisent la somme et le produit de deux rationnels. Écrire une fonction `float_of_rat` qui donne (une approximation de) la valeur décimale d'un rationnel.

Exercice 3 Définir un type `carte` (jeu de 52 cartes). Écrire une fonction construisant une carte à partir d'un couple formé d'un entier (1 donne As, 2 donne 2, ..., 13 donne roi, toute autre valeur déclenchant une erreur) et d'une couleur. Écrire une fonction qui trie un couple d'éléments. En appliquant cette fonction aux cartes, pouvez-vous déterminer comment la manière dont vous avez défini le type `carte` (et le type `couleur`) influe sur l'ordre ?

Exercice 4 On veut travailler dans le plan muni d'un repère orthonormé. Définir un type `point` correspondant à un couple de `float`. Écrire une fonction construisant un point à partir d'un couple de flottants, ainsi que des fonctions `abscisse` et `ordonnee` qui permettent d'accéder aux coordonnées d'un point.

Définir un type `vecteur` correspondant à un couple de `float` et écrire une fonction qui construit un vecteur à partir d'un couple de flottants. Écrire une fonction `addv` qui permet d'additionner deux vecteurs ainsi qu'une fonction `translate` qui permet de traduire un `point` selon un `vecteur` donné. Écrire de même une fonction qui, étant donné un couple de points A et B retourne le vecteur \vec{AB} .

L'aire d'un parallélogramme $ABCD$ est la valeur absolue de la quantité $x_1y_2 - x_2y_1$, où (x_1, y_1) est le vecteur \vec{AB} et (x_2, y_2) le vecteur \vec{AD} . Écrire une fonction qui, à partir de ces deux vecteurs, calcule l'aire du parallélogramme.

Écrire une fonction qui, à partir de trois points A, B, C calcule le point D de sorte que $ABCD$ soit un parallélogramme. Écrire une fonction qui, à partir des quatre sommets d'un parallélogramme, calcule son aire.

Exercice 5 Définir un type abstrait `ftval` qui comprend à la fois des fonctions de type `'a->'a` et des constantes. On peut combiner deux paramètres de type `ftval` soit s'il s'agit de deux fonctions de même type (auquel cas, on compose l'une avec l'autre), soit si l'un des deux paramètres est une fonction et que l'autre est une constante ; dans ce cas, on applique la fonction à la valeur (il faut que les types soient cohérents). Écrire une fonction `combine` qui effectue cette opération (si les deux paramètres sont des constantes, on retourne une erreur).

Exercice 6 Un domino contient deux valeurs comprises entre 0 (blanc) et 6. Il y a deux types de dominos : ceux qui portent deux valeurs différentes et les “doubles” qui portent deux fois la même valeur.

Évidemment, l’ordre des valeurs n’importe pas dans la définition d’un domino : le domino (3, 4) est le même que le domino (4, 3).

Définir un type abstrait **domino**.

On peut **assembler** deux dominos. Pour cela, il faut que les deux dominos aient une valeur commune. Dans ce cas, le résultat de l’assemblage est équivalent à un domino dont les deux valeurs sont celles inutilisées lors de l’assemblage : par exemple, (4, 5) s’assemble avec (2, 5) pour former (2, 4). Si les deux dominos qu’on assemble sont identiques, on choisit d’assembler les valeurs les plus petites et le résultat est évidemment un double : (2, 3) avec (2, 3) donne le double 3.

Écrire, à l’aide des opérations abstraites, une fonction qui réalise, si c’est possible, l’assemblage de deux dominos et qui, dans le cas contraire, retourne une erreur. Écrire cette même fonction à l’aide des opérations concrètes.