

Feuille 9 : Dictionnaires

Exercice 9.1 Un dictionnaire est une structure de donnée représentant un ensemble de couples (clé, valeur). La structure de donnée doit permettre :

- de consulter la valeur associée à une clé,
- d'ajouter une entrée, en associant une valeur à une nouvelle clé,
- de modifier une valeur déjà associée à une clé existante,
- de supprimer une entrée, c'est-à-dire de supprimer une clé et sa valeur associée.

Par exemple, dans un dictionnaire linguistique, les clés sont les mots et les valeurs sont les définitions. Dans la structure de données « dictionnaire », on peut bien sûr avoir des clés qui sont de type différent du type `string`, comme par exemple des clés entières.

1. Définir un type `('k, 'v) dict` permettant de représenter un dictionnaire une liste de couples.
2. Définir le dictionnaire vide `empty_dict`, de type `('k, 'v) dict`.

Exercice 9.2 Consultation on se servira du type `'a option` prédéfini par `OCaml`, que l'on peut réécrire comme :

```
type 'a option = Some of 'a | None
```

Par exemple, une valeur du type `int option` est soit `None`, soit `Some x` où `x` est un entier. L'idée est que ce type permet de représenter des valeurs optionnelles. Par exemple, `Some 3` représente l'entier 3 et `None` ne représente aucune valeur. Cela sera utile, par exemple, pour la fonction permettant de consulter la valeur associée à une clé dans un dictionnaire : si la clé n'existe pas dans le dictionnaire, la fonction retournera `None`.

Écrire une fonction `find` de type `'k -> ('k, 'v) dict -> 'v option` telle que `find key dict` calcule la valeur à la clé `key` dans le dictionnaire `dict` si cette clé existe, et `None` sinon.

Exercice 9.3 Adjonction Écrire la fonction `add` de type `'k -> 'v -> ('k, 'v) dict -> ('k, 'v) dict`, telle que `add key value dict` renvoie le dictionnaire `dict` dans lequel on a ajouté le couple `(key, value)`. Si la clé `key` n'était pas dans le dictionnaire, l'appel `add key value dict` ajoute cette clé. Sinon, elle remplace la valeur anciennement associée à `key` par `value`.

Exercice 9.4 Suppression Écrire une fonction `remove` de type `'k -> ('k, 'v) dict -> ('k, 'v) dict` telle que `remove k dict` renvoie le dictionnaire `dict` dans lequel on a enlevé la l'entrée correspondant à la clé `key`, si cette clé est présente dans `dict` et renvoie `dict` s'il ne contient pas la clé `key`.

Exercice 9.5 1. Évaluer la complexité de la fonction `find`.

2. Comment pourrait-on réduire cette complexité ?