Le type abstrait de tableau associatif

La structure de données abstraite de **tableau associatif**, appelée ici `map`, permet d'associer des éléments d'un ensemble A à des éléments d'un ensemble B. Il s'agit d'une application partielle entre les deux ensembles.

Le type tableau associatif est représenté ici par :

('a, 'b) map
'a : le type des éléments de l'ensemble A
'b : le type des éléments de l'ensemble B

Formellement, on associe à tout d : ('a, 'b) map une relation fonctionnelle $R(d) \subset A \times B$, c'est-à-dire un ensemble de couples vérifiant $(x,y) \in R(d)$ et $(x,y') \in R(d) \implies y = y'$.

Une réalisation fonctionnelle d'un tableau associatif aurait les opérations suivantes :

fonction	type	documentation
Opérations caractéristiques du tableau associatif		
INSERTION	('a, 'b) map → 'a →'b → ('a, 'b) map	ajoute au tableau associatif d'un élément de type 'a associé à la valeur de type 'b
RECHERCHE	('a, 'b) map → 'a → 'b	renvoie la valeur de type 'b associée à la valeur de type 'a donnée dans le tableau associatif
SUPPRESSION	('a, 'b) map → 'a → ('a, 'b) map	met à jour la valeur de l'élément à l'indice i du tableau t en lui assignant la valeur v
Opération complémentaire		
TABLEAUVIDE	('a, 'b) map	renvoie un tableau associatif vide

Les préconditions de certaines opérations sont :

Insertion la clé (élément de l'ensemble A) n'est pas déjà associée à un élément dans l'ensemble B (sinon il faut décider ce qu'il se passe : écrasement, interdiction ?)

Recherche la clé existe (sinon il faut également décider ce qu'il se passe)

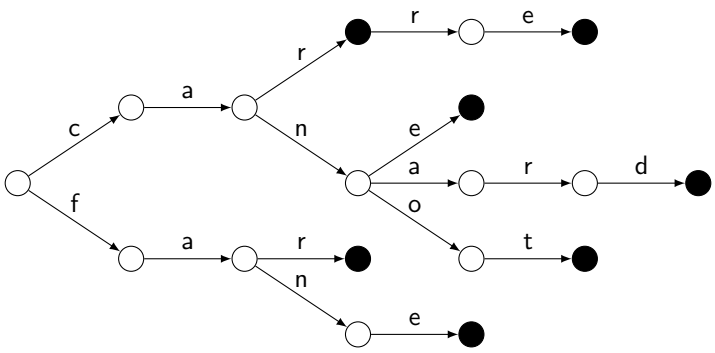
Suppression la clé existe

Avec les opérations précédentes, on demande :
- $R(\text{TableauVide}) = \emptyset$
- Recherche $m\ x = y$ si $(x,y) \in R(m)$ (y est nécessairement unique)
- $R(\text{Suppression}\ m\ x) = R(m) \setminus \{(x,y)\}$
- $R(\text{Insertion}\ m\ x\ y) = R(m) \cup \{(x,y)\}$

Il y a plusieurs manières de réaliser un tableau associatif.

Représentation du dictionnaire

Pour ce TP, vous aurez besoin de réaliser un dictionnaire de mots. Le dictionnaire permettra de déterminer si une chaîne de caractères donnée existe.
Les opérations qu'il faudra réaliser sont l'insertion et la recherche d'un mot. La suppression ne sera pas effectuée par mot : c'est l'ensemble du dictionnaire qui sera supprimé, et la mémoire associée libérée.
Nous utiliserons la représentation illustrée dans la figure ci-dessous. Un **mot** est une succession de caractères. Une succession de caractères est obtenue en partant du cercle noir le plus à gauche et en choisissant une flèche à suivre à chaque nœud. Un **mot valide**, est un mot existe réellement dans le dictionnaire français utilisé. Un mot valide est une succession de caractères qui aboutit à un **nœud terminal**, représenté par un rond rempli en noir. Les mots valides dans l'exemple sont : car, cane, carre, canard, canot, far, fane.



Réalisation du dictionnaire

Dans la langue française, les caractères autorisés sont les 26 lettres de l'alphabet, mais également certaines lettres accentuées, le tiret et l'apostrophe.

IMPORTANT 1

> Réaliser en premier lieu et valider une version qui utilise uniquement les 26 lettres minuscules de l'alphabet, sans tiret ni apostrophe, et sans accent.

Par ailleurs, les mots conservés dans le dictionnaire fourni (`fr-reforme1990_court_sans.dic`) sont uniquement écrits en minuscule.
La structure que vous devez réaliser devra consister en une liste chaînée de tableaux. Chaque élément de la structure représentera un nœud et devra offrir la possibilité de suivre l'une des lettres permettant d'atteindre un mot valide, ce que vous devrez pouvoir déterminer, y compris lorsqu'un mot valide est un préfixe d'un autre mot valide.
Ainsi, cette structure permettra de réaliser une application partielle de l'ensemble des successions de caractères à l'ensemble des booléens.

A faire pour la partie structure

Un des objectifs du projet est de réaliser en langage `C` un module `dictionnaire` réalisant la structure de données décrite ci-dessus de manière impérative.
Il faudra :

- définir la structure à utiliser ;
- implémenter des opérations `dictionnaire_vide`, `ajouter_mot`, `chercher_mot`, `supprimer_dictionnaire` ;
- utiliser tout ceci dans le programme principal, voir spécifications sur la page du projet.

Sources

- *Introduction to Algorithms, Cormen, Leiserson et Rivest ;*
- *Informatique 2021-2022, Bianquis*