TD3 : Arbres lexicographiques

Un arbre lexicographique (ou "trie") est une structure utilisée pour représenter des ensembles dont les éléments admettent une décomposition séquentielle en "caractères" possédant de plus un ordre (par exemple, les entiers admettent une décomposition en suite de chiffres, les chaînes de caractères en suite de caractères, etc).

Un arbre lexicographique contient :

- une fonction de décomposition permettant de transformer un élément en liste de "caractères" ;
- une fonction de recomposition, inverse de la précédente ;
- un arbre n-aire dont les branches représentent les suites constituant les éléments stockés dans l'ensemble, les "caractères" étant rangés dans les branches.

De plus, les branches sont triées de gauche à droite. Pour chaque nœud, il doit être indiqué si la suite lue depuis la racine constitue la décomposition d'un élément ou seulement un préfixe strict d'une telle décomposition.

Il vous est demandé de ranger **les "caractères" dans les branches**. Un exemple d'arbre lexicographique stockant le long des branches les mots du mini-dictionnaire suivant : *bas, bât, de, la, lai, laid, lait, lard, le, les, long* est présenté Fig. 1.

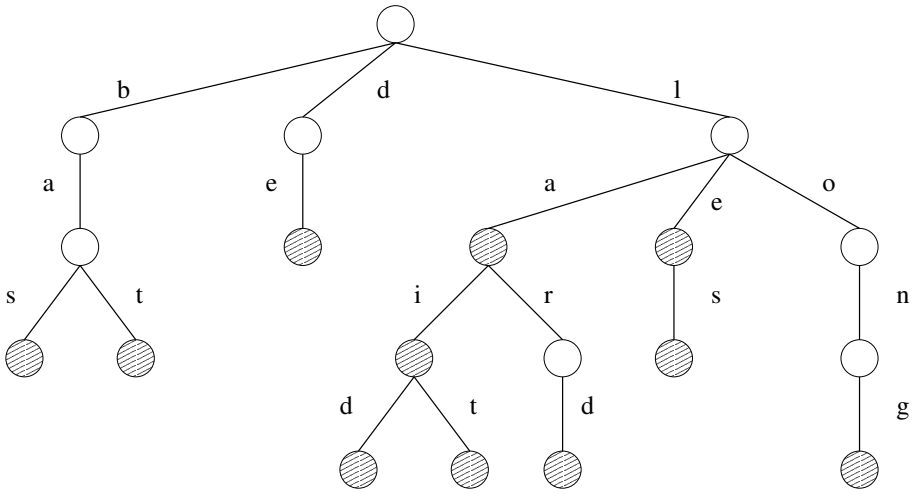

Figure 1 – Un mini-dictionnaire.

1 Structure arborescente n-aire

Dans un premier temps, nous nous intéresserons juste à la structure arborescente n-aire.

▷ **Exercice 1 (Définition des types)** *Définir le type* 'a arbre *représentant la structure arborescente : arbre n-aire avec les booléens dans les nœuds et des* 'a *dans les branches. Vous pouvez également définir un second type représentant les branches.*

▷ **Exercice 2 (Test d'appartenance)** *Écrire la fonction* appartient *qui teste si un élément appartient bien à un ensemble représenté par un arbre lexicographique.*

▷ **Exercice 3 (Ajout)** *Écrire la fonction* ajout *qui ajoute un élément à un ensemble représenté par un arbre lexicographique.*

2 Arbres lexicographiques

En plus de la structure arborescente, un arbre lexicographique contient une fonction de décomposition et une fonction de recomposition.

Par exemple, dans le cas d'un dictionnaire de mot, la fonction de décomposition sera de type `string -> char list` et la fonction de recomposition de `char list -> string`, permettant par exemple de passer de `"laid"` à `['l';'a';'i';'d']`.

▷ **Exercice 4** *Définir le type* `('a,'b) trie` *permettant de représenter des arbres lexicographiques (arbre+fonction de décomposition+fonction de recomposition).*

▷ **Exercice 5 (Test d'appartenance)** *Écrire la fonction* `appartient_trie` *qui teste si un élément appartient bien à un ensemble représenté par un arbre lexicographique.*

▷ **Exercice 6 (Ajout)** *Écrire la fonction* `ajout_trie` *qui ajoute un élément à un ensemble représenté par un arbre lexicographique.*