Travaux pratiques

option informatique

Représentation arborescente d'un dictionnaire

Les mots d'un dictionnaire peuvent être représentés par un arbre en faisant en sorte que les préfixes communs à plusieurs mots apparaissent une seule fois :

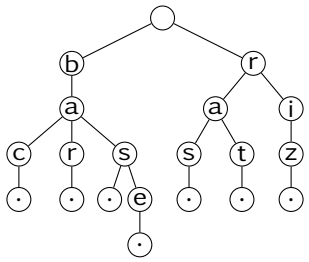

FIGURE 1 – Les mots représentés sont : bac, bar, bas, base, ras, rat, riz.

Quel est l'intérêt de faire terminer chaque mot par un point ?

On choisit de représenter un tel arbre à l'aide du type :

```
type dict = Nil | Noeud of char * dict * dict ;;
```

La première composante d'un **Noeud** contient le caractère associé à ce nœud, la seconde composante permet d'accéder au *fils ainé* de ce nœud, et la troisième composante au *frère cadet* de ce nœud.

Question 1. Dessiner le dictionnaire associé à l'arbre ci-dessus.

Question 2. Rédiger une fonction **chercher** qui détermine si un mot appartient à un dictionnaire donné.

```
chercher : string -> dict -> bool
```

Question 3. Rédiger une fonction **branche** qui prend pour argument un mot m et retourne pour résultat l'arbre réduit à la branche qui code le mot m.

```
branche : string -> dict
```

Par exemple, au mot bac sera associé l'arbre : 

Question 4. Définir une fonction **inserer** qui prend pour arguments un mot m et un arbre a et qui retourne l'arbre auquel le mot m a été ajouté (sans ordre particulier).

```
inserer : string -> dict -> dict
```

Question 5. Rédiger une fonction **creer** qui prend pour argument une liste de mots et qui retourne le dictionnaire créé à l'aide de ces mots.

```
creer : string list -> dict
```

Question 6. Rédiger enfin une fonction **extraire** qui renvoie la liste des mots contenus dans un dictionnaire donné.

```
extraire : dict -> string list
```