Travaux pratiques

Bord maximal d'un mot

Dans ce document on adopte le formalisme suivant :

- A désigne un alphabet contenant au moins deux lettres ; il sera représenté par le type *char*.
- A^* est l'ensemble des *mots* sur A, c'est-à-dire des suites finies (éventuellement vides) de lettres ; il sera représenté par le type *string*. Le mot vide sera noté ε.
- Si $u \in A^*$, on note $|u|$ sa *longueur*.
- Enfin, si $u \in A^*$ et $k \in [\![0, |u|-1]\!]$, on note u_k la $(k+1)^e$ lettre de u.

On appelle *bord* d'un mot non vide u un mot qui est à la fois préfixe propre et suffixe propre de u. Le *bord maximal* de u est l'unique bord de longueur maximale ; il sera noté $\beta(u)$. Par exemple, les bords du mot *ababa* sont ε, a et aba donc $\beta(ababa) = aba$.
On conviendra par la suite que $\beta(\varepsilon) = \varepsilon$.

Question 1.

a. Soit v un mot non vide. Montrer qu'il existe $k \leqslant |v|$ tel que l'ensemble $\left\{\beta(v), \beta^2(v), \dots, \beta^k(v)\right\}$ soit l'ensemble des bords de v.

b. Soit v un mot non vide de longueur k, et a une lettre. Montrer que $\beta(va)$ est le plus long des mots de l'ensemble $\left\{\varepsilon, \beta(v)a, \beta^2(v)a, \dots, \beta^k(v)a\right\}$ qui sont préfixes de v.

Étant donné un mot non vide u de longueur n, on définit les entiers $b(0), b(1), \dots, b(n)$ en posant :

$$b(0) = -1 \quad \text{et} \quad \forall k \in [\![1, n]\!],\ b(k) = |\beta(u_0 u_1 \cdots u_{k-1})|.$$

Autrement dit, $b(k)$ est la longueur du bord maximal du préfixe de longueur k de u.

c. Soit $k < n$. On considère les entiers $j_1 = b(k)$, $j_2 = b(j_1)$, ... , $j_{k+1} = b(j_k)$.
Montrer que $j_{k+1} = -1$. On peut donc considérer le plus petit entier α pour lequel $u_{j_\alpha} = u_k$ ou $j_\alpha = -1$. Montrer que $b(k+1) = j_\alpha + 1$.

Question 2. Déduire de ce qui précède une fonction **bord** qui au mot u associe le tableau des entiers $[b(0), b(1), \dots, b(n)]$.

```
bord : string -> int vect
```

Nous admettrons que le coût de cette fonction est linéaire.

Algorithme Knuth-Morris-Pratt et autres applications du tableau des bords

La notion de bord est principalement utilisée dans l'algorithme KMP (nommé ainsi d'après les initiales de ses découvreurs), algorithme dont l'objectif est de déterminer si un mot m (le *motif*) est facteur d'un autre mot s (la *source*).

Question 3. En considérant une lettre $x \in A$ qui n'appartient ni à m ni à s, expliquer comment le calcul du tableau b associé au mot $u = mxs$ peut permettre de résoudre ce problème, et en déduire la fonction Caml correspondante.

```
kmp : string -> string -> bool
```

Quel est le coût de celle-ci ?

Question 4. Deux mots u et v sont dits *conjugués* lorsqu'il existe deux mots r et s tels que $u = rs$ et $v = sr$. À l'aide de la fonction **kmp** rédiger une fonction qui détermine en temps linéaire si deux mots sont conjugués.

```
conjugue : string -> string -> bool
```

Question 5. À l'aide du tableau des bords, définir une fonction qui détermine en coût quadratique si un mot contient un facteur carré.

```
carre : string -> bool
```

Question 6. Une *période* d'un mot u est un préfixe non vide v de u tel que u soit préfixe de v^n avec $n \geqslant 1$. Par exemple, *abc* est une période de *abcabcab*.
On pose $u = vw$ avec $v \neq \varepsilon$. Montrer que v est une période de u si et seulement si w est un bord de u, et en déduire une fonction **periode** qui détermine la plus petite période de u en temps linéaire.

```
periode : string -> string
```

Question 7. Rédiger enfin une fonction qui détermine la liste de tous les préfixes d'un mot qui ont la propriété d'être des palindromes.

```
prefixe_palindrome : string -> string list
```