

## TD 1 : Recherche de chaînes de caractères et expressions rationnelles

### 1 Recherche d'expressions rationnelles

Le but de cette partie est de concevoir un algorithme permettant de trouver toutes les occurrences d'une expression rationnelle  $e$  dans les préfixes d'un texte  $T$  en un temps  $O(|e| \cdot |T|)$ . On commence par construire un automate *normalisé* reconnaissant  $e$ .

**Définition 1 (Automate normalisé)** *Un automate est dit normalisé s'il vérifie les conditions suivantes :*

1. *il existe un seul état initial, un seul état final, et ces deux états sont distincts ;*
2. *aucune transition ne va vers l'état initial ni ne sort de l'état final ;*
3. *tout état autre que l'état final est la source soit d'exactly une flèche étiquetée par une lettre, soit d'au plus deux flèches étiquetées par le mot vide  $\epsilon$ .*

**Définition 2 (Taille d'une expression rationnelle)** *La taille  $|e|$  d'une expression rationnelle  $e$  est son nombre de symboles. Plus précisément, on a :*

$$|e| = |a| = 1 \text{ pour } a \in A \quad |e + f| = |e \cdot f| = 1 + |e| + |f| \quad |e^*| = 1 + |e|$$

**Question 1 :** Montrer que pour toute expression rationnelle  $e$ , il existe un automate normalisé reconnaissant  $\mathcal{L}(e)$  et dont le nombre d'états est au plus  $2 \cdot |e|$ .

On cherche maintenant à calculer l'ensemble des états atteignables à partir de l'état initial en lisant un texte  $T$  de manière efficace, c'est-à-dire sans construire explicitement l'automate déterministe correspondant. Soit  $N$  la taille de l'automate obtenu à l'étape précédente.

**Question 2 :** Donner un algorithme  $\text{Trans}(P, a)$  qui, étant donné un ensemble d'états  $P$  et une lettre  $a$  renvoie l'ensemble des états successeur de  $P$  par  $a$  (sans utiliser d' $\epsilon$ -transitions). Cet algorithme devra s'exécuter en temps  $O(N)$ .

**Question 3 :** Donner un algorithme  $\text{Epsilon}(P)$  qui calcule en temps également  $O(N)$  l'ensemble des états atteignables par 0 ou plusieurs  $\epsilon$ -transitions à partir de l'ensemble d'états  $P$ .

**Question 4 :** En déduire un algorithme en temps  $O(|e| \cdot |T|)$  qui renvoie les préfixes de  $T$  qui appartiennent à  $\mathcal{L}(e)$ .

## 2 Automate des parties

On considère ici  $P \in \Sigma^*$  un motif de longueur  $m$  et on s'intéresse au langage  $L$  des mots ayant  $P$  pour suffixe.

**Question 5 :** Montrer que  $L$  est rationnel en exhibant un automate non-déterministe  $\mathcal{A}$  à  $m+1$  états  $Q = \{0 \dots m\}$  vérifiant :

$$\begin{aligned}\mathcal{L}(0) &= L \\ \mathcal{L}(k) &= \{P[k+1, m]\} \quad \forall 0 < k \leq m\end{aligned}$$

où  $\mathcal{L}(i)$  est le langage accepté par  $\mathcal{A}$  en prenant l'état  $i$  comme état initial.

On définit maintenant la fonction  $\pi^*$  de la façon suivante :

$$\begin{aligned}\pi^*(0) &= \{0\} \\ \pi^*(k) &= \{k\} \cup \pi^*(\pi(k)) \quad \forall 0 < k \leq m\end{aligned}$$

où  $\pi(k) = \max\{k' < k \mid P[1, k'] \sqsupseteq P[1, k]\}$ . (Rappel :  $w \sqsupseteq x$  denote  $w$  est suffixe de  $x$ .)

**Question 6 :** Montrer que  $\pi^*(k) = \{k' \leq k \mid P[1, k'] \sqsupseteq P[1, k]\}$  pour tout  $k \in \{0, \dots, m\}$ .

**Question 7 :** Soit  $\mathcal{A}'$  l'automate des parties issu de  $\mathcal{A}$ . Montrer que l'ensemble  $Q'$  des états atteignables dans  $\mathcal{A}'$  est exactement :

$$\{\pi^*(k) \mid 0 \leq k \leq m\}.$$

## 3 Période d'un mot

**Définition 3 (Période)** On dit que  $p \in \{1, \dots, |u|\}$  est une période d'un mot  $u \in \Sigma^*$  si pour tout  $\forall i \in \{1, \dots, |u| - p\}$ ,  $u[i] = u[i + p]$ . On note  $Period(u)$  la plus petite période de  $u$ .

On remarque que  $Period(u)$  existe toujours puisque  $|u|$  est considérée par définition comme une période de  $u$ .

**Question 8 :** On considère la fonction  $\pi$  associée au mot  $u$ . Montrer que  $Period(u) = |u| - \pi(|u|)$ .

**Question 9 :** Montrer que si  $p$  et  $q$  sont deux périodes d'un mot  $u$ , et si  $p + q \leq |u|$ , alors  $\text{pgcd}(p, q)$  est aussi une période de  $u$ .

**Question 10 :** Pour tout  $w$ , on note  $w[: -2]$  le mot  $w[1, |w| - 2]$  obtenu en tronquant  $w$  de ses deux derniers caractères.

Soit  $F_n$  la suite des mots de Fibonacci définie par

$$\begin{aligned}F_0 &= \epsilon \\ F_1 &= b \\ F_2 &= a \\ F_n &= F_{n-1} \cdot F_{n-2} \text{ for all } n \geq 3\end{aligned}$$

- Montrer que pour tout  $n \geq 5$ ,  $F_n[: -2] = F_{n-2} \cdot F_{n-1}[: -2]$ .
- Montrer que  $F_n[: -2]$  est préfixe de  $F_{n-1}^2$  et de  $F_{n-2}^3$  pour  $n \geq 5$ .
- En déduire que  $|F_{n-1}|$  et  $|F_{n-2}|$  sont des périodes de  $F_n[: -2]$ . Leur pgcd est-il une période ?

**Question 11 : (DM)** Proposez un algorithme en temps linéaire pour trouver le plus grand entier  $k$  tel que, étant donné un texte  $T$ , il existe un mot  $x$  tel que  $T = x^k$ . Démontrez sa correction.