TD 1 – Recherche de chaînes de caractères et expressions rationnelles

Margot Catinaud margot.catinaud@lmf.cnrs.fr
Valentin Dardilhac valentin.dardilhac87@gmail.com

On considère dans tout ce TD un alphabet fini Σ.

I 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 $T \in \Sigma^*$ en un temps $\Theta(|e| \cdot |T|)$. 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'exactement une flèche étiquetée par une lettre, soit d'au plus deux flèches étiquetées par le mot vide ε.

Définition 2: Taille d'une expression rationnelle

La taille $|e|$ d'une expression rationnelle e est son nombre de symboles. Plus précisément, on a :

$$|\varepsilon| = |a| \stackrel{\text{def}}{=} 1 \text{ pour } a \in \Sigma \qquad |e+f| = |e \cdot f| \stackrel{\text{def}}{=} 1 + |e| + |f| \qquad |e^*| = 1 + |e|.$$

Question 1 *Montrer que pour toute expression rationnelle e, il existe un automate normalisé reconnaissant le langage $\mathcal{L}(e)$ 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* `Trans`(P, a) *qui, étant donnés un ensemble d'états P et une lettre a renvoie l'ensemble des états successeur de P par a (sans utiliser d'ε-transitions). Cet algorithme devra s'exécuter en temps $\Theta(N)$.*

Question 3 *Donner un algorithme* `Epsilon`(P) *qui calcule en temps également $\Theta(N)$ l'ensemble des états atteignables par 0 ou plusieurs ε-transitions à partir de l'ensemble d'états P.*

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

II Automate des parties

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

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

$$\mathcal{L}(0) = \mathcal{L} \qquad et \qquad \forall\, k \in [\![1;m]\!],\ \mathcal{L}(k) = \big\{P[k+1:m]\big\},$$

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

On définit maintenant la fonction $\pi^* : [\![0;m]\!] \longrightarrow \mathcal{P}\big([\![0;m]\!]\big)$ de la façon suivante :

$$\pi^*(0) \stackrel{\text{def}}{=} \{0\} \qquad \text{et} \qquad \forall\, k \in [\![1;m]\!],\ \pi^*(k) \stackrel{\text{def}}{=} \{k\} \cup \pi^*\big(\pi(k)\big),$$

où $\pi(k) \stackrel{\text{def}}{=} \max\Big\{k' < k \mid P[1:k'] \sqsupset P[1:k]\Big\}$ (Rappel : $w \sqsupset x$ denote w est suffixe de x.)

Question 6 *Montrer que, pour tout $k \in [\![0;m]\!]$, $\pi^*(k) = \Big\{k' \leqslant k \mid P[1:k'] \sqsubset P[1:k]\Big\}$.*

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

$$\mathcal{Q}' \stackrel{def}{=} \Big\{\pi^*(k) \mid k \in [\![0;m]\!]\Big\}.$$

III Période d'un mot

Définition 3: Période

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

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

Question 8 *On considère la fonction π associée au mot u. Montrer que l'on a $\mathsf{Period}(u) = |u| - \pi(|u|)$.*

Question 9 *Soient $p, q \in [\![1;|u|]\!]$ deux périodes d'un mot $u \in \Sigma^*$ telles que $p+q \leqslant |u|$. Montrer que $\mathsf{pgcd}(p,q)$ est aussi une période de u.*

Pour tout mot $w \in \Sigma^*$, on note $w[:-2]$ le mot $w[1:|w|-2]$ obtenu en tronquant w de ses deux derniers caractères. Soit $(F_n)_{n\in\mathbb{N}}$ la suite des mots de Fibonacci définie par :

$$\forall\, n \in \mathbb{N},\ F_n \stackrel{\text{def}}{=} \begin{cases} \varepsilon & \textbf{si } n = 0, \\ b \in \Sigma & \textbf{si } n = 1, \\ a \in \Sigma & \textbf{si } n = 2, \\ F_{n-1} \cdot F_{n-2} & \textbf{sinon.} \end{cases}$$

Question 10

1. *Calculer $\mathsf{pgcd}(|F_n|, |F_{n+1}|)$ pour tout entier $n \in \mathbb{N}$.*
2. *Montrer que $F_n[:-2]$ est préfixe de F_{n-1}^2 et de F_{n-2}^3 pour tout $n \in \mathbb{N}$, $n \geqslant 5$.*
3. *En déduire que $|F_{n-1}|$ et $|F_{n-2}|$ sont des périodes de $F_n[:-2]$. Leur PGCD est-il également une période ?*

Devoir maison 1 *Proposer un algorithme en temps linéaire pour trouver le plus grand entier $n \in \mathbb{N}$ tel que, étant donné un texte $T \in \Sigma^*$, il existe un mot $u \in \Sigma^*$ tel que $T = u^n$. Démontrer sa correction.*