TD 09 – Automates farcis

Exercice 1. *BS*

Soit L un langage sur un alphabet Σ. Quelques notations :

prefixe : $x \in \mathcal{P}(L) \quad \Leftrightarrow \quad x \in \Sigma$ et $x\Sigma^* \cap L \neq \emptyset$

suffixe : $x \in \mathcal{S}(L) \quad \Leftrightarrow \quad x \in \Sigma$ et $\Sigma^* x \cap L \neq \emptyset$

facteur : $u \in \mathcal{F}(L) \quad \Leftrightarrow \quad u \in \Sigma^2$ et $\Sigma^* u \Sigma^* \cap L \neq \emptyset$

non facteur : $\mathcal{N}(L) = \Sigma^2 \setminus Fact(L)$

Langages Locaux.

Un langage L sur Σ est dit *local* si $L \setminus \{\epsilon\} = (\mathcal{P}(L)\Sigma^* \cap \Sigma^*\mathcal{S}(L)) \setminus \Sigma^*\mathcal{N}(L)\Sigma^*$.

1. Montrer que tout langage local est rationnel.

2. $(abc)^*$ est-il local ? et a^*ba ?

3. Les langages locaux sont-ils stables par union ? par concaténation ?

Un morphisme strictement alphabétique est une application $\varphi : X^* \to Y^*$ telle que

- $\varphi(uv) = \varphi(u)\varphi(v)$
- pour tout $a \in \Sigma$, $|\varphi(a)| = 1$

4. Montrer que tout langage rationnel L est l'image d'un langage local par un morphisme strictement alphabétique. (indication : on pourra utiliser un AFD ou un AFND qui reconnait L et utiliser l'alphabet $Q \times \Sigma \times Q$)

Automates locaux.

Un AFD $\mathcal{A} = (Q, i, F, \delta)$ est dit *local* si $\forall a, \exists q$ tq $\forall q', \delta(q', a) = q$ ou est indéfini [1].

5. Montrer que tout langage local sur un alphabet Σ est reconnu par un automate local standard, dont l'ensemble des états est $\Sigma \cup \{\epsilon\}$.

6. Réciproquement, montrer que tout langage reconnu par un automate local est lui-même local.

7. Soit L_1 et L_2 deux langages locaux sur des alphabets X et Y disjoints. Montrer que $L1 \cup L_2$, $L_1 \cdot L_2$ et L_1^* sont des langages locaux.

Algorithme de Berry-Sethi

une expression rationnelle e sur un alphabet Σ est dite *linéaire* si chaque lettre de Σ possède au plus une occurrence dans e.

8. Êtes-vous convaincus qu'on peut construire une version linéaire d'une expression rationnelle très facilement si l'on est autorisé à modifier l'alphabet ?

9. Soit e une expression rationnelle linéaire. Montrer que $L(e)$ est un langage local.

10. Que pensez-vous de la réciproque ?

11. Définir par induction structurelle :
- $\lambda : e \mapsto \{\epsilon\} \cap L(e)$
- $P : e \mapsto \mathcal{P}(L(e))$
- $S : e \mapsto \mathcal{S}(L(e))$
- $F : e \mapsto \mathcal{F}(L(e))$

1. C'est à dire que les transitions étiquetées par une lettre a donnée arrivent toutes dans un même état, qui ne dépend donc que de a.

12. Voici l'algorithme de Berry-Sethi qui permet d'associer à une expression rationnelle e un automate fini reconnaissant $L(e)$:
    (a) Construire une version linéaire e' de e, en mémorisant l'encodage alphabétique ;
    (b) Construire $P(e')$, $S(e')$ et $F(e')$;
    (c) Construire un AFD $\mathcal{A}'$ reconnaissant $L(e')$;
    (d) Décoder les étiquettes des transitions de $\mathcal{A}'$ pour obtenir un AFD reconnaissant $L(e)$.
13. Appliquer Berry-Sethi à $((ab(ac)^* + ca)^*b)^*$.
14. Comprendre la phrase suivante, qui conclut l'article dont est tiré le DM de taupe dont est tiré ce TD[2] :
    "Berry and Sethi have given an unusual proof of a well-known result, namely that every rational language is the homomorphic image of a local language."

Exercice 2. *KMP*

On s'intéresse à la recherche d'un motif m dans un texte t. On notera ℓ la longueur de m, et n_t celle de t. Etant donné un mot w, on notera $w[i..j]$ le sous-mot $w_i \dots w_j$ (par convention, $w[i..j] = \varepsilon$ si $i > j$).

1. Donner un algorithme naïf résolvant ce problème. Quelle est sa complexité ?

Soit $A = \langle Q, q_o, F, \Sigma, \delta \rangle$ un automate fini déterministe, et ϕ sa fonction d'état final ($\phi(w) \in Q$ est l'état dans lequel on arrive en partant de l'état initial et en lisant le mot w). On dit que A est un automate de recherche de m lorsque $Q = \{0, 1, \dots, \ell\}$ et

$$\phi(w) = \max\{q \mid m[1..q] \text{ est suffixe de } w\}$$

2. Donner l'automate de recherche du motif *abac*.
3. En supposant qu'on dispose d'un automate de recherche de m, donner un algorithme de recherche de m dans un texte. Justifier sa correction, et donner sa complexité.
4. Soit $\sigma(w) = \max\{k : m[1..k] \text{ est un suffixe de } w\}$. Montrer que $\delta(q, a) = \sigma(m[1..q] \cdot a)$.
5. Montrer que pour tout automate de recherche associé au motif m, sur l'entrée t, $\phi(t[1..i]) = \sigma(t[1..i])$ pour tout i. **Indication.** On pourra montrer que si $q = \sigma(w)$, alors $\sigma(w \cdot a) = \sigma(m[1..q] \cdot a)$.
6. En déduire un algorithme de calcul d'un automate de recherche de m en $O(\ell^3)$

Nous allons maintenant essayer d'améliorer cette complexité. Un *bord* d'un mot w est un sous-mot strict de w qui soit à la fois son préfixe et son suffixe. On définit $\pi(q)$, pour $q \leq \ell$, comme étant la taille du plus grand bord de $m[1..q]$.

7. Que vaut $\delta(q, a)$ si $m_{q+1} = a$? Si $q = m$ ou $m_{q+1} \neq a$, montrer que $\delta(q, a) = \delta(\pi(q), a)$.
   En supposant la fonction π connue, en déduire un algorithme linéaire de calcul de l'automate de recherche.
8. Calculer les valeurs de π pour $m = abcababcabcabb$.
9. Montrer que $\pi(q) > 0 \implies \exists i,\ \pi(q) = \pi^i(q-1) + 1$.
   En déduire une définition inductive de $\pi(q)$.
10. Trouver un algorithme linéaire calculant la fonction de suffixe.
11. Vous venez de re-découvrir l'algorithme de *Knuth-Morris-Pratt*. Quelle est sa complexité ?

2. cosigné par Jean BERSTEL et Jean-Éric PIN