TD 2 – Automates finis et expressions rationnelles

1 Langages locaux

Soit Σ un alphabet. On dit qu'un langage L sur Σ est *local* lorsqu'il existe $P, S \subseteq \Sigma$ et $N \subseteq \Sigma^2$ tels que $L \setminus \{\varepsilon\} = (P\Sigma^* \cap \Sigma^* S) \setminus \Sigma^* N \Sigma^*$.

Exercice 1.

Caractériser la classe des langages locaux en termes d'AFD, c'est-à-dire, définir une classe d'AFD dits *locaux* telle qu'un langage est local si et seulement s'il est reconnu par un AFD local.

Exercice 2.

Soit L_1 un langage local sur un alphabet Σ_1 et L_2 un langage local sur un alphabet Σ_2 tels que $\Sigma_1 \cap \Sigma_2 = \emptyset$. Montrer que $L_1 \cup L_2$ et $L_1 L_2$ sont locaux.

Exercice 3.

Soit L un langage local sur un alphabet Σ. Montrer que L^* est local.

2 Algorithme de Glushkov

Une expression rationnelle sur un alphabet Σ est dite *linéaire* lorsqu'elle contient au plus une fois chaque lettre $a \in \Sigma$.

Exercice 4.

Montrer que toute expression rationnelle linéaire correspond à un langage local.

Exercice 5.

Étant donné une expression rationnelle linéaire E sur un alphabet Σ, donner un algorithme permettant de déterminer $P, S \subseteq \Sigma$ et $N \subseteq \Sigma^2$ tels que $\mathcal{L}(E) \setminus \{\varepsilon\} = (P\Sigma^* \cap \Sigma^* S) \setminus \Sigma^* N \Sigma^*$.

Exercice 6.

En déduire un algorithme permettant, étant donné une expression rationnelle E, de construire un AFN reconnaissant $\mathcal{L}(E)$.

Exercice 7.

Utiliser cet algorithme pour obtenir un AFN reconnaissant le langage correspondant à $(a(ab)^*)^* + (ba)^*$ sur $\{a, b\}$.

3 Équations linéaires

Exercice 8. *Lemme d'Arden*

Soit K et L deux langages sur un alphabet Σ. Montrer que si $\varepsilon \notin K$, alors il existe un unique $X \subseteq \Sigma^*$ tel que $X = KX \cup L$ en explicitant X.

Exercice 9.

Qu'advient-il du lemme précédent si K contient le mot vide ?

Exercice 10.

Étant donné $n \in \mathbb{N}_{>0}$, soit $K_{i,j}$ pour $i, j \in [\![1, n]\!]$ ainsi que L_i pour $i \in [\![1, n]\!]$ des langages sur un alphabet Σ. Montrer que si, pour tous $i, j \in [\![1, n]\!]$, $\varepsilon \notin K_{i,j}$, alors il existe $X_1, X_2, \ldots, X_n \subseteq \Sigma^*$ uniques tels que

$$\begin{aligned} X_1 &= K_{1,1}X_1 \cup K_{1,2}X_2 \cup \cdots \cup K_{1,n}X_n \cup L_1 \\ X_2 &= K_{2,1}X_1 \cup K_{2,2}X_2 \cup \cdots \cup K_{2,n}X_n \cup L_2 \\ &\vdots \qquad\qquad\qquad \vdots \\ X_n &= K_{n,1}X_1 \cup K_{n,2}X_2 \cup \cdots \cup K_{n,n}X_n \cup L_n \ , \end{aligned}$$

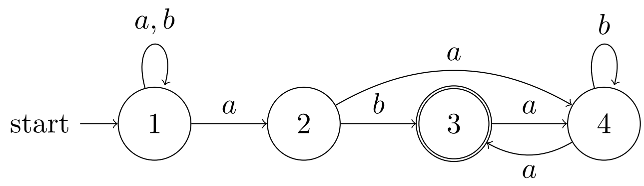
qui de plus sont rationnels dès que $K_{i,j}$ pour $i, j \in [\![1, n]\!]$ et L_i pour $i \in [\![1, n]\!]$ le sont.

Exercice 11.

En déduire un algorithme permettant, étant donné un AFN $\mathcal{A}$, de construire une expression rationnelle correspondant à $\mathcal{L}(\mathcal{A})$.

Exercice 12.

Donner une expression rationnelle correspondant au langage reconnu par l'AFN suivant, sur $\{a, b\}$:



.

4 Caractérisations

Soit Σ un alphabet.

Exercice 13.

Donner une condition nécessaire et suffisante pour qu'un AFD $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ vérifie que $\mathcal{L}(\mathcal{A})$ est une combinaison booléenne de langages de la forme $\{w\}$ avec $w \in \Sigma^*$.

Exercice 14.

Donner une condition nécessaire et suffisante sur la forme d'une expression rationnelle E sur Σ pour que $\mathcal{L}(E)$ soit reconnu par un AFD $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ ne contenant aucun circuit non trivial (c'est-à-dire, tel qu'il n'existe pas $u, v \in \Sigma^*$ et $q, p \in Q, p \neq q$ vérifiant $\hat{\delta}(q, u) = p$ et $\hat{\delta}(p, v) = q$).

Exercice 15.

Donner une condition nécessaire et suffisante sur la forme d'une expression rationnelle E sur Σ pour que $\mathcal{L}(E)$ soit reconnu par un AFD $A = (Q, \Sigma, \delta, q_0, F)$ tel que pour tous $u, v \in \Sigma^*$ et $p, q \in Q$, si $\hat{\delta}(q, uv) = p$, alors $\hat{\delta}(q, vu) = p$.

Exercice 16.

Donner une condition nécessaire et suffisante pour qu'un AFD $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ vérifie que $\mathcal{L}(\mathcal{A})$ est une combinaison booléenne de langages de la forme $\Sigma^* a \Sigma^*$ avec $a \in \Sigma$.