

## 1 Rappels : les langages suivants sont-ils rationnels ? (Justifiez)

**Exercice 1.**  $\mathcal{L}_0 = \{w \in \{a, b, c\}^* \mid (|w|_a = 0) \Rightarrow (|w|_b = 0)\}$

**Solution 1.**  $c^* + (b + c)^* a(a + b + c)^*$

**Exercice 2.**  $\mathcal{L}_2 = \{w \in \{a, b\}^* \mid |w|_a < |w|_b\}$

**Solution 2.** Non. Sinon on aurait un  $n$  de pompage et sur le mot  $a^n b^{2n}$  on aurait  $a^{n+kx} b^{2n}$  dans le langage pour  $k > 0$  et tout  $x$ . Or  $kx + n > 2n$  pour  $x$  grand.

**Exercice 3.**  $\mathcal{L}_1 = \{w \in \{a, b\}^* \mid 7 \text{ divise } |w|_a, 3 \text{ divise } |w|_b, \}$

**Solution 3.** Oui, c'est l'intersection de  $b^*((ab^*)^7)^*$  et de  $a^*((ba^*)^3)^*$

**Exercice 4.**  $\mathcal{L}_3 = \{w \in \{(\cdot)\}^* \mid w \text{ est bien parenthésé}\}$

**Solution 4.** Non. Sinon on aurait un  $n$  de pompage et sur le mot  $(\cdot)^n$  on aurait  $(\cdot)^{n+kx}$  pour  $k > 0$  et tout  $x$  dans le langage.

## 2 Non équivalence des lemmes de pompages

Voici trois lemmes de pompage pour un langage  $\mathcal{L}$  :

- 1  $\exists n \in \mathbb{N}, \forall u \in \mathcal{L} : |u| \geq n \Rightarrow \exists v, t, w \in \Sigma^* \quad u = vt w \quad |t| > 0 \quad \forall m \in \mathbb{N} \quad vt^m w \in \mathcal{L}$
- 2  $\exists n \in \mathbb{N}, \forall r u s \in \mathcal{L} : |u| \geq n \Rightarrow \exists v, t, w \in \Sigma^* \quad u = vt w \quad |t| > 0 \quad \forall m \in \mathbb{N} \quad r v t^m w s \in \mathcal{L}$
- 3  $\exists n \in \mathbb{N}, \forall r u_1 \dots u_n s \in \mathcal{L}, \forall i : |u_i| \geq 1 \Rightarrow \exists 1 \leq i < j \leq n \quad \forall m \in \mathbb{N} \quad r u_1 \dots u_{i-1} (u_i \dots u_j)^m u_{j+1} \dots u_n s \in \mathcal{L}$

**Exercice 5.** Montrer que  $\mathcal{L} = \{w \in \{a, b\}^* \mid |w|_a = |w|_b\}$  vérifie le lemme 1 mais pas le lemme 2.

**Solution 5.** Pour tout mot de taille supérieure à 1 on peut trouver une alternance  $a, b$  et on peut la répéter. Supposons qu'il existe  $n$ , on pose  $r = a^n, s = \epsilon, u = b^n$  et alors  $u$  ne peut pas être découpé comme il faudrait.

**Exercice 6.** Montrer que  $\mathcal{L} = \{(ab)^n (cd)^n \mid n \in \mathbb{N}\} \cup \Sigma^* (aa + bb + cc + dd + ac + bd) \Sigma^*$  vérifie le lemme 2 mais pas le lemme 3.

**Solution 6.** Soit  $r u s \in \mathcal{L}_\Sigma$  avec  $|u| \geq 5$ .

Soit  $r u s \in \Sigma^* aa + bb + cc + dd + ac + bd \Sigma^*$  et  $r u s = w x y v$  avec  $w, v$  des mots et  $x, y$  une paire de lettre de  $aa + bb + cc + dd + ac + bd$ . Comme  $|u| \geq 3$  alors  $u$  contient une lettre qui n'est pas  $x$  ou  $y$  et peut être retirée ou multipliée.

Soit  $r u s = (ab)^k (cd)^k$  et donc  $u$  contient  $bab$  ou  $cdc$  en prenant  $t = a$  ou  $t = d$  on obtient le résultat attendu car on a, soit un symbole  $bb$  ou  $cc$  quand  $m = 0$ , soit le mot original quand  $m = 1$  soit  $aa$  ou  $dd$  quand  $m \geq 2$ .

Soit  $n \in \mathbb{N}^*$ . On pose  $r = \epsilon, u_i = (ab)$  et  $s = (cd)^n$  et le lemme 3 ne tient plus, car  $(ab)^{n+(j-i-1) \times k} (cd)^n$  n'appartient pas au langage pour  $k \neq 1$ .

## 3 Algorithme de Brzowski

Soit  $\mathcal{L}_\Sigma$  un langage régulier sur l'alphabet  $\Sigma$  et soit  $\mathcal{A} = \langle \mathcal{Q}, \Sigma, \delta, i, \mathcal{F} \rangle$  un automate déterministe qui reconnaît  $\mathcal{L}_\Sigma$  tel que tous les états de  $\mathcal{A}$  soient accessibles depuis  $i$ .

**Exercice 7.** Montrer que  $Rev(\mathcal{L}_\Sigma) = \{a_n \dots a_1 \mid a_1 \dots a_n \in \mathcal{L}_\Sigma\}$  est reconnu par l'automate (non déterministe)  $Rev(\mathcal{A}) = \langle \mathcal{Q}, \Sigma, \delta', \mathcal{F}, \{i\} \rangle$  où  $v \in \delta'(q, c) \Leftrightarrow \delta(v, c) = q$ .

**Solution 7.** Étant donné un mot  $v_1 \dots v_n$ , tout parcours  $q_1, q_2, \dots, q_n$  dans l'automate  $\mathcal{A}$  est équivalent à un parcours inversé  $q_n, \dots, q_2, q_1$  dans  $\mathcal{A}'$  (i.e.  $\tilde{\delta}(q_1, v_1 \dots v_i) = q_i \Leftrightarrow \tilde{\delta}'(q_i, v_i \dots v_1) = q_1$ ). Un parcours est acceptant dans  $\mathcal{A}$  si et seulement si son renversé est acceptant dans  $\mathcal{A}'$  ( $q_1 = i$  et  $q_n \in \mathcal{F}$ ). Donc  $Rev(\mathcal{A})$  accepte bien  $Rev(\mathcal{L}_\Sigma)$ .

**Definition 1.** Étant donné un automate non déterministe  $\mathcal{B} = \langle \mathcal{Q}_\mathcal{B}, \Sigma, \delta_\mathcal{B}, \mathcal{I}_\mathcal{B}, \mathcal{F}_\mathcal{B} \rangle$  On note  $Det(\mathcal{B})$  l'automate obtenu par la construction par sous-ensemble. On a  $Det(\mathcal{B}) = \langle 2^{\mathcal{Q}_\mathcal{B}}, \Sigma, 2^{\delta_\mathcal{B}}, \mathcal{I}_\mathcal{B}, \{q \in 2^{\mathcal{Q}_\mathcal{B}} \mid q \cap \mathcal{F}_\mathcal{B} \neq \emptyset\} \rangle$  avec  $2^{\delta_\mathcal{B}}(e, v) = \bigcup_{q \in e} \delta_\mathcal{B}(q, v)$ .

**Exercice 8.** Montrer que dans  $Det(\mathcal{B}) = \langle 2^{\mathcal{Q}_\mathcal{B}}, \Sigma, 2^{\delta_\mathcal{B}}, \mathcal{I}_\mathcal{B}, \{q \in 2^{\mathcal{Q}_\mathcal{B}} \mid q \cap \mathcal{F}_\mathcal{B} \neq \emptyset\} \rangle$  on a :  $q \in 2^{\tilde{\delta}_\mathcal{B}}(\mathcal{I}_\mathcal{B}, w_1 \dots w_n)$  équivalent à l'existence de  $q_1, \dots, q_n, q_{n+1}$  avec  $q_1 \in \mathcal{I}_\mathcal{B}$ ,  $q_{n+1} = q$  et  $q_{i+1} \in \delta_\mathcal{B}(q_i, w_i)$ .

**Solution 8.** Par récurrence sur  $n$ , montrons l'existence de  $q_1 \dots q_{n+1}$  :

- pour  $n = 0$ ,  $q_1 = q$  convient ;
- pour  $n > 0$  on a  $q \in 2^{\tilde{\delta}_\mathcal{B}}(\mathcal{I}_\mathcal{B}, w_1 \dots w_n w_{n+1})$  qui nous donne par définition de  $2^{\delta_\mathcal{B}}$ ,  $q_{n+1} \in 2^{\tilde{\delta}_\mathcal{B}}(\mathcal{I}_\mathcal{B}, w_1 \dots w_n)$  tel que  $q \in \delta_\mathcal{B}(q_{n+1}, w_{n+1})$ . On trouve alors  $q_1 \dots q_{n+1}$  par récurrence et  $q_1 \dots q_{n+1} q_{n+2}$  avec  $q_{n+2} = q$  convient.

Maintenant s'il existe  $q_1 \dots q_{n+1}$ , alors par itération sur  $i$  que  $q_{i+1} \in 2^{\tilde{\delta}_\mathcal{B}}(\mathcal{I}_\mathcal{B}, w_1 \dots w_i)$  on a :

- $q_1 \in \mathcal{I}$  et donc  $q_1 \in 2^{\tilde{\delta}_\mathcal{B}}(\mathcal{I}_\mathcal{B}, \epsilon)$
- $2^{\tilde{\delta}_\mathcal{B}}(\mathcal{I}_\mathcal{B}, w_1 \dots w_{i+1}) = \bigcup_{e \in 2^{\delta_\mathcal{B}}(\mathcal{I}_\mathcal{B}, w_1 \dots w_i)} \delta_\mathcal{B}(e, w_{i+1})$  or  $q_i \in 2^{\tilde{\delta}_\mathcal{B}}(\mathcal{I}_\mathcal{B}, w_1 \dots w_i)$  donc  $q_{i+1} \in 2^{\tilde{\delta}_\mathcal{B}}(\mathcal{I}_\mathcal{B}, w_1 \dots w_{i+1})$

**Definition 2.** On pose  $leftEquiv(x, y) = \forall z : (zx \in \mathcal{L}_\Sigma) \Leftrightarrow (zy \in \mathcal{L}_\Sigma)$

**Exercice 9.** Soit  $Det(Rev(\mathcal{A})) = \langle \mathcal{Q}_d, \Sigma, \delta_d, i_d, \mathcal{F}_d \rangle$ . Montrer que pour tout  $x, y \in (\Sigma^*)^2$  on a :

$$LeftEquiv(x, y) \Rightarrow (\tilde{\delta}_d(i_d, rev(x)) = \tilde{\delta}_d(i_d, rev(y)))$$

**Solution 9.**

Soient  $x$  et  $y$  tels que  $Q_x = \delta_d(i_d, x) \neq \delta_d(i_d, y) = Q_y$ . On a  $Q_x \setminus Q_y \neq \emptyset$  ou  $Q_y \setminus Q_x \neq \emptyset$  et donc par symétrie entre  $x$  et  $y$  on peut choisir  $q \in Q_x \setminus Q_y$ . Comme tous les états de  $\mathcal{A}$  sont accessibles, on a  $z$  tel que  $\tilde{\delta}(i, z) = q$ .

Soit  $w \in \{x, y\}$  avec  $w = w_1 \dots w_n$ . Un mot  $zw = zw_1 \dots w_l$  est dans  $\mathcal{L}_\Sigma$  quand  $\tilde{\delta}(i, zw) = \tilde{\delta}(q, w) \in \mathcal{F}$  et donc il existe  $q_1, \dots, q_{l+1}$  avec  $q_1 = q$ ,  $q_{l+1} \in \mathcal{F}$  et  $q_i = \tilde{\delta}(q, w_1 \dots w_i)$ . D'après l'exercice ??, un tel parcours n'existe que pour  $x$  car il correspondrait à un parcours renversé dans  $Det(Rev(\mathcal{L}_\Sigma))$ .

**Exercice 10.** En déduire à l'aide du théorème de Myhill–Nerode que  $Det(Rev(\mathcal{L}_\Sigma))$  est l'automate minimal reconnaissant  $Rev(\mathcal{L}_\Sigma)$ .

*Rappel : Étant donné un langage  $\mathcal{L}$  on peut regarder la relation d'équivalence  $RightEquiv(x, y) = \forall z : xz \in \mathcal{L} \Leftrightarrow yz \in \mathcal{L}$ . Le théorème de Myhill–Nerode dit que tout automate reconnaissant le langage contient au moins autant d'états que le nombre de classes d'équivalence.*

**Exercice 11.** En déduire une construction de l'automate minimal reconnaissant  $\mathcal{L}_\Sigma$ .

**Solution 11.**  $Det(Rev(Det(Rev(\mathcal{A}))))$  est minimal et reconnaît  $Rev(Rev(\mathcal{L}_\Sigma)) = \mathcal{L}_\Sigma$ .

**Exercice 12.** Quelle est la complexité de cette construction ?

**Solution 12.** L'automate minimal résultant est forcément plus petit que l'automate mais la taille de l'automate déterministe (minimal) du langage renversé peut exploser (au pire  $2^n$ ). Par exemple pour :

