Langages et automates

OPTION INFORMATIQUE - Devoir n° 4 - Olivier Reynet

A Automates

A1. En exhibant un automate et sans justification particulière, montrer que le langage constitué par les entiers binaires pairs est reconnaissable.

A2. Soit l'alphabet des chiffres en base dix : $\Sigma = \{0,1,2,3,4,5,6,7,8,9\}$. En exhibant un automate et sans justification particulière, montrer que le langage des multiples de 1000 en base 10 est reconnaissable.

> ■ **Définition 1 — Fonction de transition étendue aux mots (rappel).** La fonction de transition peut être étendue aux mots par passages successifs d'un état à un autre en lisant les lettres d'un mot.
>
> On définit inductivement cette fonction étendue noté δ^* :
>
> $$\forall q \in Q, \delta^*(q,\epsilon) = q \tag{1}$$
>
> $$\forall q \in Q, \forall w \in \Sigma^*, \forall a \in \Sigma, \delta^*(q, w.a) = \delta(\delta^*(q,w),a) \tag{2}$$

A3. Montrer que l'opération de complétion d'un automate est correcte, c'est-à-dire que $\mathcal{L}_{rec}(C(\mathcal{A})) = \mathcal{L}_{rec}(\mathcal{A})$, si $C(\mathcal{A})$ est l'automate complété de $\mathcal{A}$.

A4. Soit Σ un alphabet et $\mathcal{A}$ l'automate fini déterministe défini par $\mathcal{A} = \left(Q, \Sigma, q_i, \delta, \{q_f\}\right)$. Pour cet automate, on suppose que pour chaque symbole s de Σ, on a $\delta(q_i, s) = \delta(q_f, s)$.

(a) En procédant par induction structurelle sur les mots, démontrer la propriété $\mathcal{P} : \forall u \in \Sigma^* \setminus \{\varepsilon\}, \delta^*(q_i, u) = \delta^*(q_f, u)$.

(b) En procédant par récurrence sur k, démontrer la propriété

$$\mathcal{P}_k : \forall u \in \Sigma^* \setminus \{\varepsilon\}, \left(u \in \mathcal{L}_{rec}(A) \Rightarrow \forall k \in \mathbb{N}^\star, u^k \in \mathcal{L}_{rec}(A)\right)$$

A5. Un automate fini déterministe $\mathcal{A}$ est défini par $\mathcal{A} = \left(Q, \Sigma, q_i, \delta, F\right)$.

(a) Proposer un algorithme COMPLEMENTAIRE($\mathcal{A}$) qui construit l'automate complémentaire de $\mathcal{A}$. Ne pas implémenter l'algorithme, le décrire.

(b) Proposer un algorithme ACCESSIBLES($\mathcal{A}$) qui renvoie l'ensemble des états accessibles d'un automate $\mathcal{A}$. Ne pas implémenter l'algorithme, le décrire.

(c) Proposer un algorithme EST_VIDE($\mathcal{A}$) qui permet de savoir si le langage reconnu par un automate est vide. La donnée d'entrée est de type automate, la valeur renvoyée est booléenne. Ne pas implémenter l'algorithme, le décrire. Il est possible de se servir des algorithmes précédents.

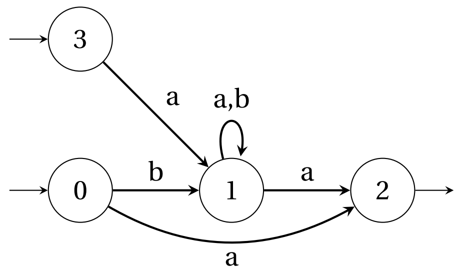
On considère deux langages réguliers $\mathcal{L}$ et $\mathcal{L}'$ et leurs automates finis déterministes associés $\mathcal{A}$ et $\mathcal{A}'$.

(d) Proposer une relation équivalente à $\mathcal{L} \subseteq \mathcal{L}'$, en utilisant l'intersection et la complétentation.

(e) En déduire un algorithme INCLUS_DANS($\mathcal{A}$,$\mathcal{A}'$)permettant de déterminer si $\mathcal{L} \subseteq \mathcal{L}'$.

(f) En déduire un algorithme EGAL($\mathcal{A}$,$\mathcal{A}'$) permettant de déterminer si $\mathcal{L} = \mathcal{L}'$.

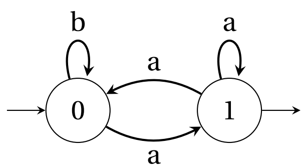
A6. Soit l'expression régulière $e = aba^*(ba|ca)$. En utilisant l'algorithme de Berry-Sethi, construire l'automate de Glushkov associé à e.



A7. Déterminiser l'automate suivant :

A8. En utilisant la méthode compositionnelle (automates de Thompson), construire un automate non déterministe avec transitions spontanées qui reconnaît le langage dénoté par $a^*(b|c)$.

A9. En utilisant la méthode de l'élimination des états, calculer une expression régulière dont le langage dénote le langage reconnu par l'automate suivant.



B Langages

B1. Soit L un langage. Montrer que les propositions suivantes sont équivalentes :

(i) $\varepsilon \in L$

(ii) $\forall i \geq 0, \varepsilon \in L^i$

(iii) $\forall i \geq 0, L^i \subseteq L^{i+1}$

B2. On considère un alphabet $\Sigma = \{a, b\}$. Pour les ensembles suivant, trouver l'expression régulière qui le dénote.

(a) L'ensemble des mots de longueur paire.

(b) L'ensemble des mots contenant un nombre impair de a. Par exemple : ab ou baabbbab.

(c) L'ensemble des mots de composés des motifs aa ou bb ou du mot vide. Par exemple : bbaaaabb.

B3. Simplifier, en justifiant, les expressions régulières suivantes :

(a) $\varepsilon|ab|abab|(ab)^*$

(b) $a(ab^*|aa)|aa(b^*|a)$

B4. Le langage $\mathcal{L} = \{a^p b^q, (p, q) \in \mathbb{N}^2\}$ est-il régulier ?

B5. Montrer que le langage L défini par $L = \{baba^2ba^3 \dots ba^n, n \in \mathbb{N}^\star\}$ n'est pas régulier.

> ■ **Définition 2 — Mot miroir.** Soit Σ un alphabet et w un mot sur Σ. Soient $a_1, \dots, a_n$ des lettres de Σ. Le mot miroir d'un mot $w = a_1 a_2 \dots a_n$ est $w^R = a_n a_{n-1} \dots a_1$.

B6. Soit $\Sigma = \{a, b\}$, un alphabet. En utilisant un langage proche du langage des puissances, montrer que $\mathcal{L} = \{w \in \Sigma^*, w = w^R\}$ n'est pas régulier.

B7. Soient x, y, z des mots sur un alphabet Σ. Montrer que :

$$x^2 = y^2 z^2 \Leftrightarrow \exists w \in \Sigma^*, x = yz \wedge (x, y, z) \in \{w\}^*$$

C Minimalisation

On considère l'automate fini déterministe $\mathcal{A}$ défini comme suit :

- **Alphabet** : $\Sigma = \{a, b\}$,
- **Ensemble des états** : $Q = \{0, 1, 2, 3\}$,
- **État initial** : 0,
- **Ensemble des états acceptants** : $F = \{2, 3\}$,
- **Fonction de transition** (δ) représentée par la table suivante :

État actuel	a	b
0	1	0
1	1	2
2	3	2
3	3	2

C1. Dessiner l'automate correspondant à cette représentation tabulaire.

Soit q un état d'un automate A.

On note $L_q = \{w \in \Sigma^*, \delta^*(q, w) \in F\}$.

On peut alors définir une relation d'équivalence, l'équivalence de Nérode, entre états notée $\sim_A$ définie par :

$$q \sim_A p \Leftrightarrow L_q = L_p \tag{3}$$

C2. Montrer que la relation $\sim_A$ est bien une relation d'équivalence.

On cherche maintenant à contruire l'automate minimal $\tilde{\mathcal{A}}$ associé à $\mathcal{A}$.

Les classes d'équivalence sont les états de l'automate minimal. On les notera $\{q_j, \ldots, q_k\}$, d'après les états qui les composent.

L'état initial de $\tilde{\mathcal{A}}$ est la classe contenant l'état initial de l'automate $\mathcal{A}$.

Une classe $\{q_j, \ldots, q_k\}$ est acceptante si elle contient au moins un état acceptant de l'automate original $\{q_j, \ldots, q_k\} \cap F \neq \emptyset$.

Pour créer les classes d'équivalence, on choisit d'abord une première partition (arbitraire) des états de l'automate en choisissant d'un côté l'ensemble des états accepteurs et de l'autre les autres états. Pour trouver les classes d'équivalence, on construit un tableau représentant les transistions depuis les parties. Si, pour une même partie, une lettre fait transiter dans deux parties différentes, alors on affine la partie en la scindant en deux. On recommence jusqu'à obtenir des classes d'équivalence.

C3. Établir les classes d'équivalence de Nérode pour l'automate $\mathcal{A}$.

C4. Dessiner l'automate minimal $\tilde{\mathcal{A}}$.