

Le but de ce document est d'être un sujet d'*entraînement*. Il devrait être réalisable en 1h ou 1h30, mais estimer la durée d'un sujet est un problème complexe. Quelques questions de code OCaml sont présentes dans le sujet.

Ce sujet se divise en trois parties *quasi-indépendantes*.

1 Quelques questions de cours.

Q1. Déterminiser l'automate représenté en figure 1.

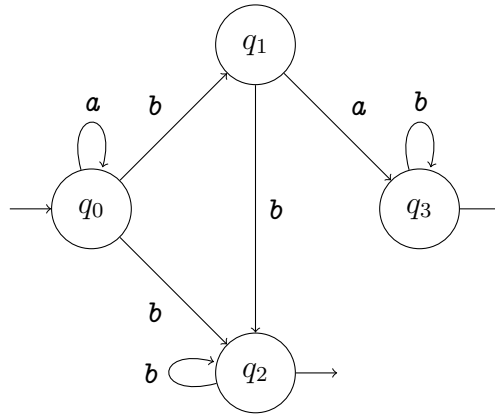


Figure 1 | Un automate exemple \mathcal{A} .

Q2. On fixe un alphabet $\Sigma = \{a, b\}$. Construire un automate déterministe reconnaissant le langage des mots contenant un nombre n de ‘ a ’, où $n \equiv 3 \pmod{5}$.

Q3. On pose $\Sigma = \{0, 1\}$. Le langage

$$L = \left\{ 0^{n_1} 1^{m_1} 0^{n_2} 1^{m_2} \dots 0^{n_k} 1^{m_k} \mid \begin{array}{l} k \geq 0 \\ \forall i \in \llbracket 1, k \rrbracket, n_i > 0 \\ \forall j \in \llbracket 1, k \rrbracket, m_j > 0 \end{array} \right\}$$

est-il régulier ? Justifier.

2 Automates produits.

L'objectif des automates produits est de pouvoir prouver quelques propriétés de clôture sur les langages reconnaissables.

2.1 Fonction de transition.

- Q4.** Étant donné un automate non déterministe \mathcal{A} , quelle(s) condition(s) doit(/doivent) vérifier \mathcal{A} pour qu'il soit déterministe ?
- Q5.** Démontrer que l'ensemble des transitions δ d'un automate déterministe et complet est équivalent à une fonction de la forme $\delta' : Q \times \Sigma \rightarrow Q$.
- Q6.** Écrire un type `nfa` (*non-deterministic finite automaton*) représentant un automate fini non déterministe. On pourra représenter les états par des entiers et les lettres de Σ par des caractères.
- Q7.** Similairement à la question 6, définir un type `dfa` (*deterministic finite automaton*) qui représente un automate fini déterministe et complet. On représentera l'ensemble des transitions δ par la fonction de transition (*c.f.* question 5)
- Q8.** Écrire une fonction OCaml `dfa_of_nfa : nfa -> dfa` qui transforme un automate fini déterministe représenté par le type `nfa` en sa représentation avec le type `dfa`. On s'assurera que l'automate est bien déterministe. S'il ne l'est pas, on renverra une exception

2.2 Construction de l'automate.

On considère deux automates déterministes \mathcal{A} et \mathcal{B} , de même alphabet Σ fini.

On construit l'automate $\mathcal{C} = (Q_{\mathcal{C}}, \Sigma, \delta_{\mathcal{C}}, I_{\mathcal{C}}, F_{\mathcal{C}})$ par

- ▷ $Q_{\mathcal{C}} = Q_{\mathcal{A}} \times Q_{\mathcal{B}}$;
- ▷ $I_{\mathcal{C}} = I_{\mathcal{A}} \times I_{\mathcal{B}}$;
- ▷ $F_{\mathcal{C}} = F_{\mathcal{A}} \times F_{\mathcal{B}}$;
- ▷ $\delta_{\mathcal{C}}((p, q), \sigma) = (\delta_{\mathcal{A}}(p, \sigma), \delta_{\mathcal{B}}(q, \sigma))$ pour $(p, q) \in Q_{\mathcal{C}}$ et $\sigma \in \Sigma$.

Q9. Démontrer que l'automate \mathcal{C} est déterministe.

Q10. Démontrer que $\mathcal{L}(\mathcal{C}) = \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{B})$ par double inclusion.

L'automate \mathcal{C} est appelé « *automate produit* » : il est similaire à un produit cartésien des automates \mathcal{A} et \mathcal{B} (hormis pour la fonction de transition δ).

Q11. En modifiant la construction de l'automate \mathcal{C} , construire un automate \mathcal{D} déterministe dont le langage est $\mathcal{L}(\mathcal{A}) \triangle \mathcal{L}(\mathcal{B})$, où \triangle désigne la *différence symétrique* ensembliste.

3 Stabilité par mélange parfait.

On définit une opération, appelée *mélange parfait*. Le *mélange parfait* de deux langages A et B , avec $A, B \subseteq \Sigma^*$, est

$$A \sqcup B = \left\{ a_1 b_1 a_2 b_2 \dots a_n b_n \mid \begin{array}{l} a = a_1 a_2 \dots a_n \in A \\ b = b_1 b_2 \dots b_n \in B \end{array} \right\},$$

où chaque a_i et b_j (pour $i \in \llbracket 1, n \rrbracket$) est un élément de Σ .

Q12. Démontrer que la classe des langages réguliers est close sous l'opération « mélange parfait ».

Indication 1.

On pourra partir de deux automates \mathcal{A} et \mathcal{B} reconnaissant les langages A et B respectivement, et construire un automate reconnaissant $A \sqcup B$.

Indication 2.

Repenser à la section 2, et réaliser une construction similaire.