

TD n°3

Automates : produit, nondéterminisme

Exercice 1 (Digicode) On veut écrire deux automates déterministes et complets qui reconnaissent le « mot de passe » d’un digicode. Les seules entrées possibles sont les chiffres, et le code est 1165.

1. Construire un automate qui accepte séquence tapée qui finit par le bon code.
2. Construire un automate qui lit un code de taille 4, l’accepte uniquement si c’est le bon, mais permet ensuite de retenter sa chance.

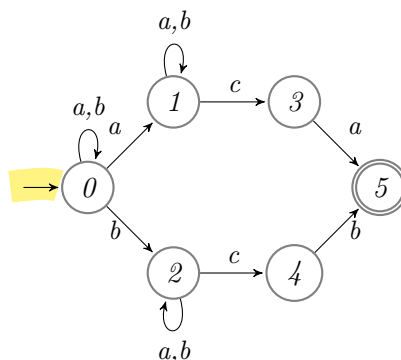
Exercice 2 (Produit d’automates) Pour deux automates déterministes et complets $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$ et $\mathcal{A}' = (\Sigma, Q', q'_0, F', \delta')$, on définit l’automate déterministe $\mathcal{A}'' = (\Sigma, Q \times Q', (q_0, q'_0), F'', \delta'')$, dit automate produit, avec $\delta''((q, q'), a) = (\delta(q, a), \delta'(q', a))$, et l’ensemble F'' des états acceptants dépend de ce que l’on veut calculer.

1. Dessiner un automate \mathcal{A}_1 déterministe et complet qui reconnaît le langage \mathcal{L}_1 des mots sur $\{a, b\}$ qui commencent par a . (3 états devraient suffire.)
2. Dessiner un automate \mathcal{A}_2 déterministe et complet qui reconnaît le langage \mathcal{L}_2 des mots sur $\{a, b\}$ qui finissent par b . (2 états devraient suffire.)
3. Dessiner le produit des deux automates \mathcal{A}_1 et \mathcal{A}_2 , sans s’occuper des états acceptants. Éliminer le(s) état(s) non accessible(s) éventuel(s). (On dit qu’un état est accessible si on peut atteindre cet état en lisant un mot depuis un état initial.)
4. Comment choisir les états acceptants pour obtenir :

$$(a) \quad \mathcal{L}_1 \cap \mathcal{L}_2, \quad (b) \quad \mathcal{L}_1 \cup \mathcal{L}_2, \quad (c) \quad \mathcal{L}_1 \setminus \mathcal{L}_2, \quad (d) \quad \overline{\mathcal{L}_1 \cap \mathcal{L}_2}.$$

5. (*) Pour lesquels de ces calculs était-il possible d’utiliser des automates déterministes non complets pour \mathcal{A}_1 et \mathcal{A}_2 ?

Exercice 3 (Automates non déterministes) Soit \mathcal{A}_1 l’automate fini non déterministe sur l’alphabet $\{a, b, c\}$ ci-dessous :

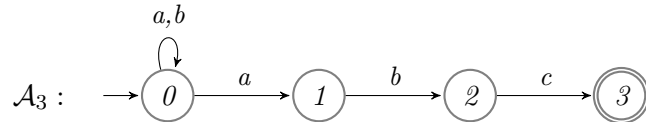
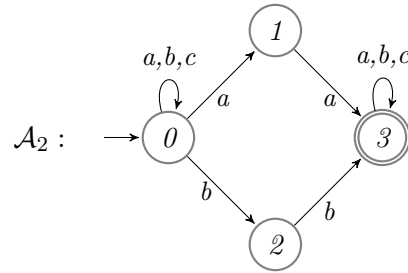


1. Parmi les mots suivants, lesquels sont acceptés par l’automate \mathcal{A}_1 ?

☐ abca ☐ abaacb ☐ bcba ☐ aaacb ☐ ca

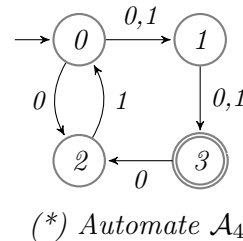
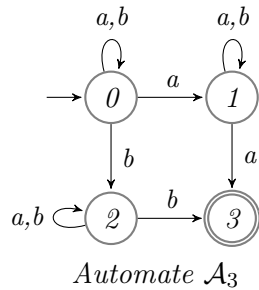
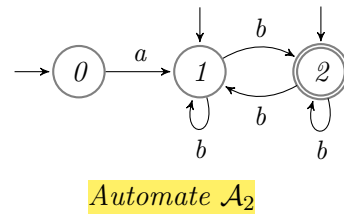
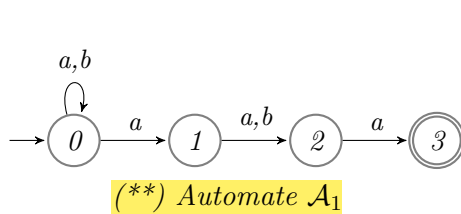
2. Donner une expression rationnelle définissant le langage accepté par \mathcal{A}_1 .

3. Même question pour les automates suivants



4. (*) Donner un automate non déterministe à trois états acceptant le langage $a^* + (ab)^*$.

Exercice 4 (Déterminisation) Déterminiser les automates suivants :



Exercice 5 (Complémentaire, Miroir) Pour commencer, on veut déterminer si les langages constitués de tous les mots qui ne contiennent pas un motif donné sont reconnaissables. En particulier, on s'intéressera au langage $\mathcal{L} = \{u \in \{a, b, c\}^* \mid u \text{ ne contient pas le facteur } aba\}$.

1. Donner un automate \mathcal{A}_1 non déterministe pour $\mathcal{L}_1 = \{u \in \{a, b, c\}^* \mid u \text{ contient le facteur } aba\}$.
2. Donner un automate \mathcal{A}_2 déterministe pour \mathcal{L}_1 .
3. Rappeler comment construire un automate reconnaissant le langage complémentaire d'un autre automate. L'appliquer à \mathcal{A}_2 pour obtenir \mathcal{A}_3 reconnaissant \mathcal{L} .
4. Peut-on généraliser ce raisonnement et conclure que le complémentaire d'un langage reconnaissable est toujours reconnaissable ?
5. (*) Si $u = x_0 \dots x_{n-1}$ est un mot, on définit le miroir de ce mot par $\tilde{u} = x_{n-1} \dots x_0$. On définit le langage miroir de \mathcal{M} par $\tilde{\mathcal{M}} = \{\tilde{u} \mid u \in \mathcal{M}\}$. Donner un algorithme général qui permet de construire un automate reconnaissant le langage miroir d'un autre automate. Faut-il que l'automate de départ soit déterministe ?
6. Appliquer cet algorithme à \mathcal{A}_2 . L'automate construit est-il déterministe ?
7. Appliquer cet algorithme à \mathcal{A}_1 . Que pouvez-vous dire de $\tilde{\mathcal{L}}$?