TD n°3

Déterminisation et Automate produit

Les exercices ou questions marqués d'une ou plusieurs étoiles sont facultatifs ; le nombre d'étoiles donne une idée de la difficulté ou de la longueur des calculs.

Déterminisation

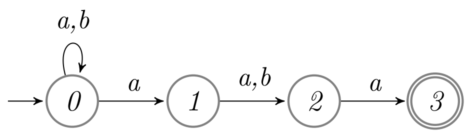
Étant donné un automate fini non déterministe (AFND) $\mathcal{A} = (\Sigma, Q, I, F, \delta)$, il est possible de construire un automate fini déterministe $\mathcal{A}' = (\Sigma, 2^Q, I, \{P \mid P \cap F \neq \emptyset\}, \delta')$ équivalent, c'est-à-dire qui accepte le même langage :

— l'alphabet est identique ;
— les états de $\mathcal{A}'$ sont les parties de l'ensemble d'états Q ;
— l'unique état initial de $\mathcal{A}'$ est l'ensemble I des états initiaux de $\mathcal{A}$;
— les états acceptants de $\mathcal{A}'$ sont ceux qui comportent au moins un état acceptant de $\mathcal{A}$;
— pour tout état $P \in 2^Q$ de $\mathcal{A}'$ (un ensemble d'états de $\mathcal{A}$) et tout symbole $a \in \Sigma$, on a :

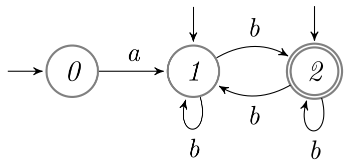
$$\delta'(P, a) = \bigcup_{q \in P} \delta(q, a).$$

L'algorithme correspondant, appelé *construction par sous-ensembles*, consiste informellement à construire les états atteignables de $\mathcal{A}'$ en calculant petit à petit le résultat de δ', en débutant à partir de l'état initial I. De cette façon, on n'obtient pas forcément tous les sous-ensembles de Q.

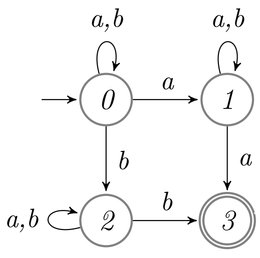
Exercice 1 *Déterminiser les automates suivants :*



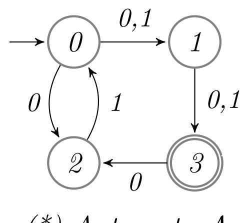
(**) Automate $\mathcal{A}_1$



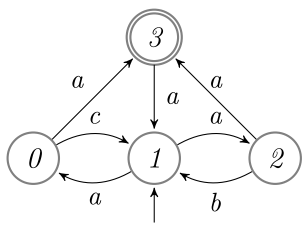
Automate $\mathcal{A}_2$



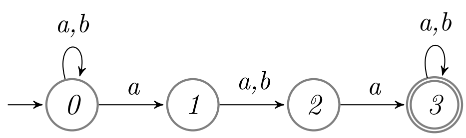
Automate $\mathcal{A}_3$



(*) Automate $\mathcal{A}_4$



Automate $\mathcal{A}_5$



(**) Automate $\mathcal{A}_6$

Exercice 2 (Digicode) *On veut écrire directement sans passer par l'algorithme de déterminisation deux automates déterministes et complets qui reconnaissent l'entrée du « mot de passe » d'un digicode. Il n'y a que des chiffres possibles en entrée. Le code est* 1165.

1. *Construire un automate qui arrive dans un état acceptant pour toute séquence tapée qui finit par le bon code.*
2. *Constuire un automate qui lit un code de taille 4, l'accepte si c'est le bon, refuse sinon, et permet ensuite de retenter sa chance.*

Exercice 3 (Complémentaire, Intersection) *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. *Donner un automate* $\mathcal{A}_3$ *déterministe pour* $\mathcal{L}$ *à partir de* $\mathcal{A}_2$.
4. *Peut-on généraliser ce raisonnement et conclure que le complémentaire d'un langage reconnaissable est toujours reconnaissable ?*
5. *En utilisant ce résultat et le fait que l'union de deux langages reconnaissables est reconnaissable (pourquoi ?), comment peut-on conclure que l'intersection de deux langages reconnaissables est toujours reconnaissable ?*

Exercice 4 ((*) Langage miroir) *Le langage miroir d'un langage* $\mathcal{L}$ *est le langage* $\widetilde{\mathcal{L}} = \{\widetilde{u} \mid u \in \mathcal{L}\}$, *avec* $\widetilde{u} = x_{n-1} \cdots x_0$ *pour* $u = x_0 \cdots x_{n-1}$. *Par exemple pour le langage*

$$\mathcal{L}_1 = \{w \in \{\flat, \sharp\}^* \mid \text{la troisième lettre de } w \text{ est un } \sharp\}.$$

on a

$$\widetilde{\mathcal{L}_1} = \{w \in \{\flat, \sharp\}^* \mid \text{la troisième lettre de } w \text{ à partir de la fin est un } \sharp\}.$$

1. *Donner une expression rationnelle pour* $\mathcal{L}_1$ *et une pour* $\widetilde{\mathcal{L}_1}$.
2. *Décrire un procédé permettant de construire en général une expression rationnelle pour* $\widetilde{\mathcal{L}}$ *à partir de l'expression rationnelle pour* $\mathcal{L}$.
3. *Décrire un procédé permettant de construire l'automate qui reconnaît le langage* $\widetilde{\mathcal{L}}$ *étant donné celui de* $\mathcal{L}$. *Est-ce qu'en commençant avec un automate déterministe pour* $\mathcal{L}$ *on obtient un automate déterministe pour* $\widetilde{\mathcal{L}}$ *?*

Automate produit

Exercice 5 (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) $\mathcal{L}_1 \cap \mathcal{L}_2$, *(b)* $\mathcal{L}_1 \cup \mathcal{L}_2$, *(c)* $\overline{\mathcal{L}_1 \cap \mathcal{L}_2}$.
5. **Bonus :** *Pour lesquels de ces calculs était-il possible d'utiliser des automates déterministes non complets pour* $\mathcal{A}_1$ *et* $\mathcal{A}_2$ *?*