Langages Formels

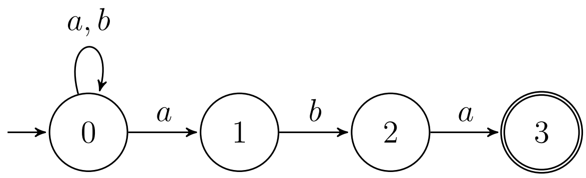
TD 2

Isa Vialard
vialard@lsv.fr

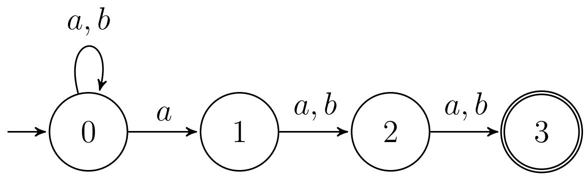
February 9, 2024

Exercise 1 : Déterminisation

1. Pour chacun des automates $\mathcal{A}_1$ et $\mathcal{A}_2$, déterminer le langage reconnu et donner un automate déterministe équivalent.



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



(b) Automate $\mathcal{A}_2$

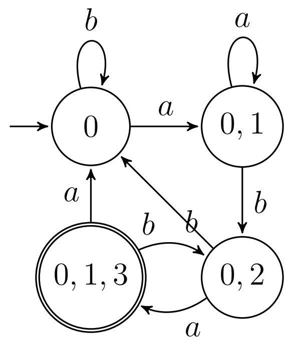
2. Si $\Sigma = \{a, b\}$, montrer que le langage $L = \Sigma^* a \Sigma^{n-1}$ ne peut être reconnu par un automate déterministe de moins de 2^n états.

   Soit $\mathcal{A}$ un automate déterministe reconnaissant L d'état initial i, et de fonction de transition étendue δ^*.

   Soient u et v deux mots distincts de Σ^n. Sans perte de généralité, il existe u', v', w tels que $u = u'aw$ et $v = v'bw$. Supposons $\delta^*(i, u) = \delta^*(i, v)$. Puisque $\delta^*(i, ua^{n-1-|w|}) = \delta^*(i, va^{n-1-|w|})$ et $ua^{n-1-|w|} \in L$, on a que $va^{n-1-|w|} \in L$ ce qui est contradictoire.

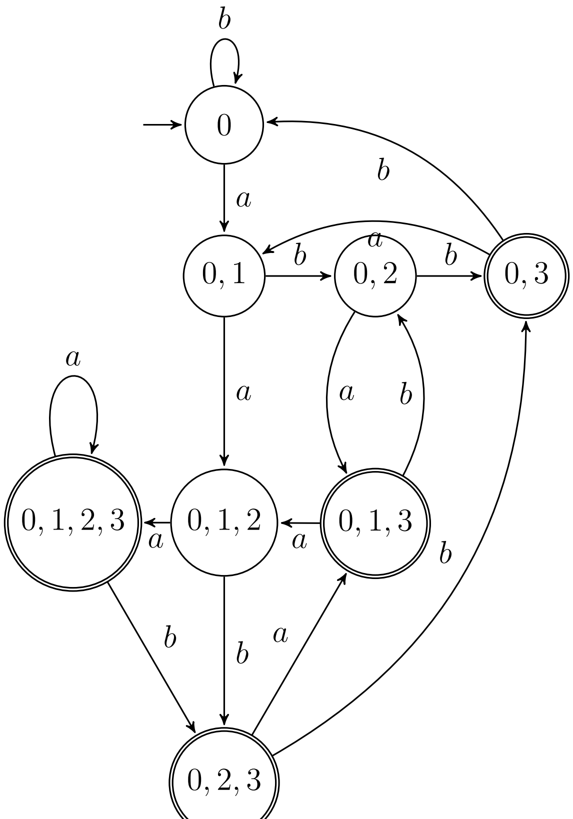
   Ainsi, il y a au moins autant d'états dans $\mathcal{A}$ que de mots de Σ^n, c'est-à-dire au moins 2^n.

3. Conclure sur le nombre d'états d'automates déterminisés par rapport à leur version non déterministe.



(a) Automate $\mathcal{A}_1$ déterminisé

Le langage reconnu est Σ^*aba.



(b) Automate $\mathcal{A}_2$ déterminisé

Le langage reconnu est $\Sigma^*a\Sigma^2$.

Par définition de l'algorithme de déterminisation, pour tout automate ND à n états on a un automate D à 2^n états ou moins. L'exemple précédent ne permet pas de montrer que cette borne est atteinte, puisqu'on a un automate ND à $n+1$ états qui se déterminise en au moins 2^n états. Pour des exemples où la borne est atteinte, voir la p.5 de `https://arxiv.`

`org/pdf/1509.03254.pdf`, ou encore `https://cs.stackexchange.com/questions/3381/nfa-with-exponential-number-of-states-when-determinized`

Exercise 2 : Des étoiles

On considère les énoncés suivants, de plus en plus forts.

Soit $L \subseteq \Sigma^*$, il existe $N > 0$ tel que pour tout $x \in L$:

(a) Si $|x| \geq N$ alors il peut s'écrire $x = u_1u_2u_3$ avec $u_2 \neq \varepsilon$ et $u_1u_2^*u_3 \subseteq L$.

(b) Si $x = w_1w_2w_3$ avec $|w_2| \geq N$ alors $w_2 = u_1u_2u_3$ avec $u_2 \neq \varepsilon$ et $w_1u_1u_2^*u_3w_3 \subseteq L$.

(c) Si $x = uv_1v_2 \dots v_M w$ avec $|v_i| \geq 1$ et $M \geq N$, alors il existe $0 \leq j < k \leq M$ tel que $uv_1 \dots v_j(v_{j+1} \dots v_k)^*v_{k+1} \dots v_M w \subseteq L$.

1. Montrer que tout langage reconnaissable vérifie l'énoncé (a).

Soit $\mathcal{A} = \langle Q, i, F, \delta \rangle$ un automate déterministe reconnaissant un langage L. Soit $N = |Q| + 1$ et $x \in L$ tel que $|x| \geq N$.

Par principe des tiroirs, il existe u_1, u_2, u_3 tels que $x = u_1u_2u_3$ et $\delta^*(i, u_1) = \delta^*(i, u_1u_2)$. Notons q cet état.

On a que $\delta^*(q, u_2) = q$, et donc par récurrence immédiate : pour tout $k \in \mathbb{N}$, $\delta^*(q, u_2^k) = q$. Nous savons de plus que $\delta^*(q, u_3) \in F$ car $x \in L$.

Donc tout mot de $u_1u_2^*u_3$ a une exécution acceptante dans $\mathcal{A}$.

2. Montrer que $\{ a^nb^n \ : \ n \in \mathbb{N} \}$ n'est pas reconnaissable.

Par l'absurde : on applique le critère (a) sur $x = a^Nb^N$, et on effectue une disjonction de cas : $u_2 = a^i$, $u_2 = a^ib^j$ ou $u_2 = b^j$. Les trois cas mènent à des absurdités.

On remarque qu'on s'embêterait moins avec le critère (b) en choisissant $w_1 = a^N$, $w_2 = b^N$, $w_3 = \varepsilon$.

On admet que les langages reconnaissables satisfont les trois propriétés (a), (b), et (c).

3. Montrer que le langage $\{ a^{n^2} \ : \ n \in \mathbb{N} \}$ n'est pas reconnaissable.

Critère (a): on prend $x = a^{N^2}$. Il existe $0 < m \leq N^2$ tel que $a^{N^2+mk} \in L$ pour tout $k \in \mathbb{N}$. Contradiction.

4. Montrer que $\{ u \ : \ u = \widetilde{u} \}$, c'est à dire le langage des palindromes, n'est pas reconnaissable.

Par l'absurde : on applique le critère (b) sur $x = a^Nba^N$ et $w_1 = a^Nb$, $w_2 = a^N$, $w_3 = \varepsilon$. On conclut à une contradiction.

5. De même pour $\{ (ab)^n(cd)^n \ : \ n \in \mathbb{N} \} \cup (\Sigma^*\{ aa, bb, cc, dd, ac \}\Sigma^*)$.

   Par l'absurde : on applique le critère (c) sur $x = (ab)^N(cd)^N$ avec $u = w = \varepsilon$, $v_1 = \cdots = v_N = ab$, et $v_{N+1} = \cdots = v_{2N} = cd$. On conclut à une contradiction de manière similaire à la question 2.

6. Montrer que le langage suivant satisfait (c) mais n'est pas reconnaissable: $\{ udv \ : \ u, v \in \{a, b, c\}^*, \ u \neq v$ ou bien l'un des deux contient un carré $\}$. On pourra admettre l'existence d'une infinité de mots sans carrés sur l'alphabet $\{a, b, c\}$.

   Supposons qu'il soit reconnaissable par un automate déterministe $\mathcal{A}$ à N états. Il existe deux mots $w_1, w_2 \in \{ a, b, c \}^*$ de taille supérieure à N que $\mathcal{A}$ ne sait différencier, qui ne contiennent pas de carrés. Or, w_1dw_2 est dans le langage mais pas w_2dw_2, ce qui est absurde.

   On peut noter $x = udv$. On prend une décomposition selon (c), avec $M \geq 4$. Au moins deux v_i adjacents sont dans u ou v, on suppose qu'ils sont dans u sans perte de généralité, et on note $u = u_1v_iv_{i+1}u_2$. On ne peut avoir $u_1v_iu_2 = u_1u_2 = v$. On choisit le bon facteur à itérer ne rendant pas les composantes égales pour 0, laissant le mot identique pour 1 et créant un carré ensuite.

Exercise 3 : Blind bartender

A blind bartender with boxer gloves plays the following game with a customer: he has a tray in front of him with four glasses arranged in a square. Each of these glasses may or may not be turned, without the bartender knowing. The purpose of the bartender is to arrange so that all the glasses are turned in the same direction. To do this, he can each turn choose one of the following three actions:

- Turn one of the glasses
- Turn two neighboring glasses
- Turn two opposing glasses

To add to the difficulty, the customer can turn the tray any number of quarter turns between each of the bartender's actions. The game ends as soon as one of the two winning positions is reached.

Show that we can restrict the number of different configurations to four, then represent the possible actions of the game by a non-deterministic automaton.

Determine this automaton and deduce a winning strategy for the bartender.

Soit $\Sigma = \{Op, N, 1\}$ l'alphabet des actions du bartender : Op retourne deux verres opposés, 1 un seul verre, N deux verres voisins.

Le plateau peut être en quatre états: G pour l'état gagnant (4 verres dans le même sens), 1 quand un verre n'est pas dans le même sens que les autres,

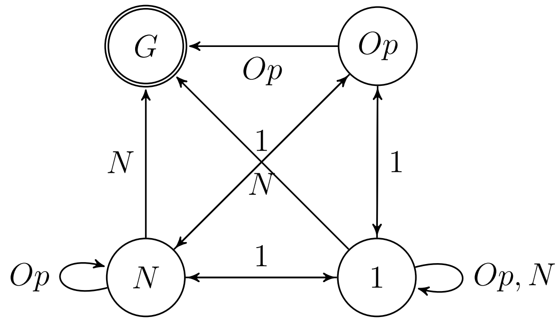


Figure 3: Automate du bartender

N quand 2 verres côte à côte sont dans un sens et les autres dans l'autre, Op quand 2 verres opposés sont dans un sens et les autres dans l'autre. Ce qui nous donne un automate non-déterministe (on ne prend pas la peine de représenter les transitions partant de G).

Si on déterminise en partant de l'état $All = 1, N, Op, G$ (le jeu est dans une des 4 configurations), le bartender peut atteindre G de manière sûre en effectuant la suite d'actions $Op, N, Op, 1, Op, N, Op$:

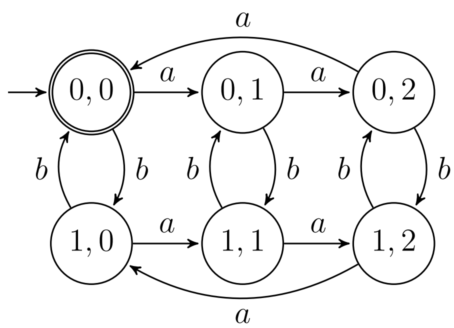
$$All \to^{Op} 1, N, G \to^{N} 1, 0, G \to^{Op} 1, G \to^{1} O, N, G \to^{Op} N, G \to^{N} O, G \to^{Op} G$$

Contrôle continu 2

À rendre pour le 08/02 au début du TD.

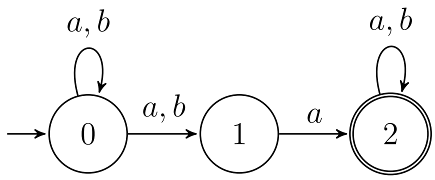
Exercise 4 :

Dessiner l'automate qui reconnait le langage des mots w sur l'alphabet $\{a, b\}$ tels que $|w|_b$ est pair et $|w|_a$ est divisible par 3.

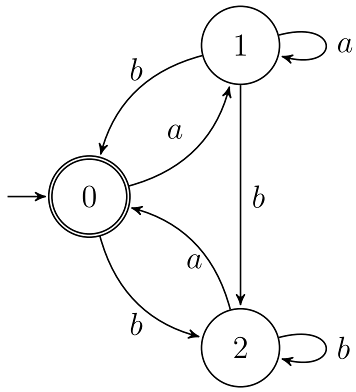


Exercise 5 : Déterminisation

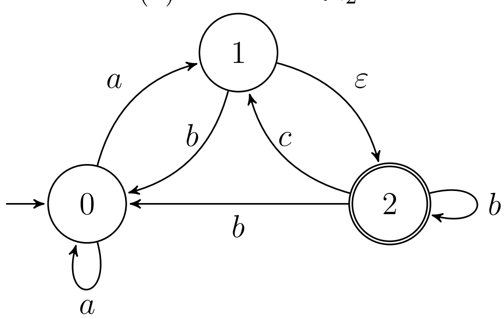
Déterminisez les automates suivants. Quels langages reconnaissent-ils ?



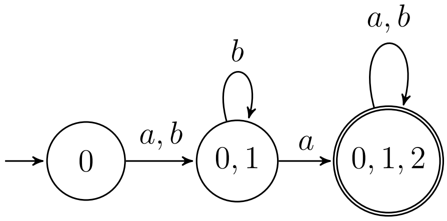
(a) Automate $\mathcal{A}_1$



(b) Automate $\mathcal{A}_2$

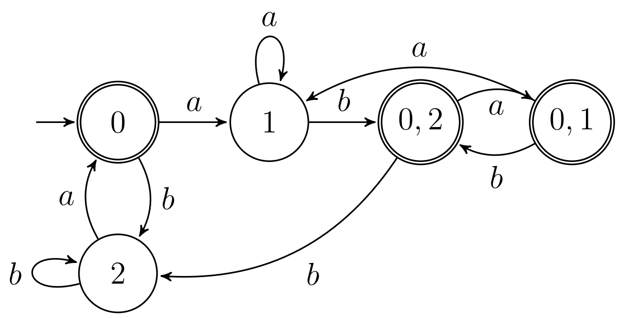


(c) Automate $\mathcal{A}_3$



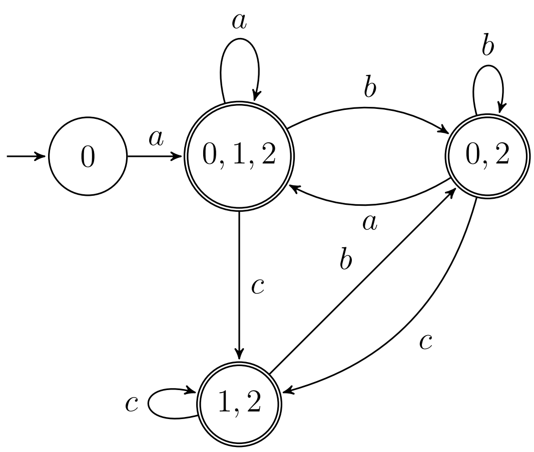
(a) Automate $\mathcal{A}_1$ déterminisé

Le langage reconnu est $\Sigma^+ a \Sigma^*$.



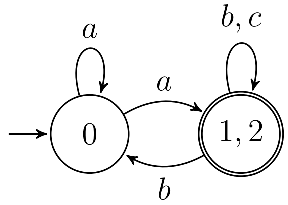
(b) Automate $\mathcal{A}_2$ déterminisé

Le langage reconnu est $(a^+b + a^+b^+a + b^+a)^*$.

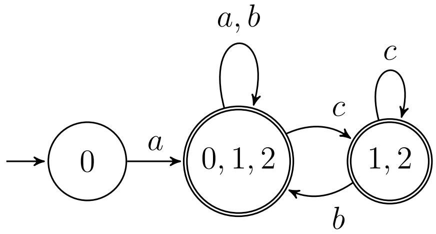


(a) Automate $\mathcal{A}_3$ déterminisé

Ça c'est si on déterminise bêtement. Sinon on peut aussi remarquer qu'on peut réduire $\mathcal{A}_3$ en fusionnant les états 1 et 2 :



(b) Automate $\mathcal{A}_3$ réduit



(c) Automate $\mathcal{A}_3$ réduit puit déterminisé

Ainsi on voit plus facilement que $\mathcal{L}(\mathcal{A}) = a^+(b + c + ba^+)$, ou encore le langage des mots qui commencent par a et ne contiennent pas le facteur ca.