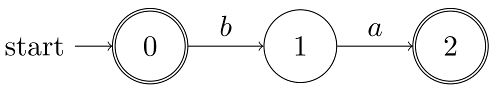
Lucas Larroque

TD 1 – Langages, automates finis (non) déterministes

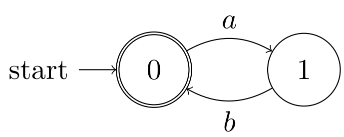
1 Automates déterministes et langages reconnus

Exercice 1. Pour chacun des automates suivants, donner le langage qu'il reconnaît.

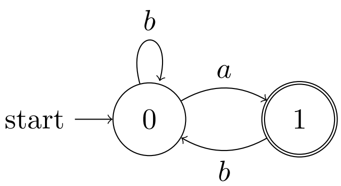
a)



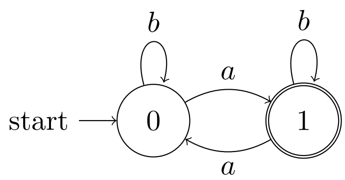
b)



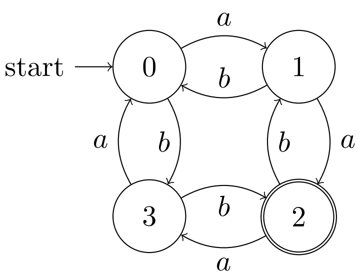
c)



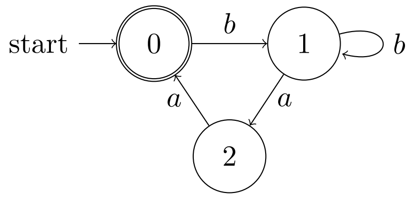
d)



e)



f)



Exercice 2. Pour chacun des langages suivants, donner un automate fini déterministe (AFD) le reconnaissant.

a) Les mots sur l'alphabet $\{a\}$ de longueur multiple de 3.

b) Pour chaque $d \in \mathbb{N}$ les mots sur l'alphabet $\{a\}$ de longueur multiple de d.

c) Les mots sur l'alphabet $\{a, b\}$ contenant un nombre pair de a et impair de b.

d) Les mots sur l'alphabet $\{a, b\}$ contenant le facteur aab ou $aaab$.

e) Les représentations binaires d'entiers pairs. Ici entier est entendu au sens positif et les nombres sont donnés dans l'ordre gros-boutiste (c'est-à-dire l'ordre normal de lecture des nombres : 1 puis 0 puis 1 puis 0 puis 1 puis 0, c'est le nombre binaire 101010 soit 42 en décimal).

f) Pour chaque $d \in \mathbb{N}$, les représentations binaires des entiers multiples de d.

g) Pour chaque $(d, c) \in \mathbb{N}^2$, les représentations binaires des entiers de la forme $c + k \cdot d$ pour $k \in \mathbb{N}$.

2 Équivalence entre automates et langages rationnels

Dans cette partie, on se propose de montrer le théorème de Kleene, sur l'équivalence entre automates et langages rationnels.

Theorème 1 (Kleene 1956)**.** *Un langage L est rationnel si et seulement s'il existe un automate fini le reconnaissant.*

On commence par montrer le sens direct par récurrence.

Exercice 3. Soient deux langages L_1 et L_2 respectivement reconnus par les automates $\mathcal{A}_1$ et $\mathcal{A}_2$. Pour chacun des langages suivants, donner un automate fini le reconnaissant.

a) $L_1 \cup L_2$ b) $L_1 \circ L_2$ c) $(L_1)^*$

Exercice 4. En conclure que si L est un langage rationnel, alors il existe un automate fini $\mathcal{A}$ tel que $\mathcal{L}(\mathcal{A}) = L$.

Pour le sens réciproque, on utilise une méthode d'élimination d'états. Cette méthode repose sur la notion d'automate généralisé, une généralisation des automates finis non déterministes dans laquelle les transitions ne sont plus étiquetées par des lettres, mais par des expressions régulières. On notera l'ensemble des expressions régulières $\mathcal{R}$.

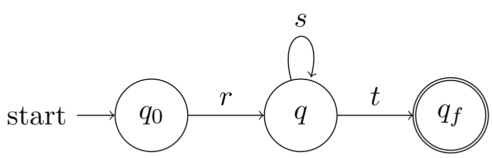
Définition 2. *Un* automate généralisé $(Q, \Sigma, q_0, q_f, \delta)$ *est la donnée de :*
- *Un ensemble fini non vide Q d'*états*,*
- *Un ensemble fini non vide Σ de lettres, appelé* alphabet*,*
- *Un état* initial $q_0 \in Q$ *et un état* final $q_f \in Q$*,*
- *Une fonction de* transition $\delta : (Q \setminus \{q_f\}) \times (Q \setminus \{q_0\}) \to \mathcal{R}$.

Un tel automate accepte un mot w s'il existe $w_1, \ldots, w_n$ et $q_1, \ldots, q_n$ tels que $w = w_1 \ldots w_n$, $q_n = q_f$, et pour tout $i \leq n-1$, $w_i \in \mathcal{L}(\delta(q_i, q_{i+1}))$.

Exercice 5. Montrer qu'à tout automate on peut associer un automate généralisé qui reconnaît le même langage.

L'idée de l'algorithme est ensuite d'éliminer les états un à un en mettant à jour la fonction de transition à chaque fois.

Exercice 6. Donner le langage reconnu par l'automate généralisé $\mathcal{A} = (\{q_1, q, q_2\}, \Sigma, q_1, q_2, \delta)$ où $\delta(q_1, q) = r$, $\delta(q, q) = s$ et $\delta(q, q_2) = t$, représenté ci-contre.



Exercice 7. Soient $\mathcal{A} = (Q, \Sigma, q_0, q_f, \delta)$ un automate généralisé et q un état de cet automate différent de q_0 et q_f. Pour toute paire d'états $q_1, q_2 \in Q \setminus \{q\}$, on pose e_{q_1,q_2} l'expression rationnelle représentant le langage trouvé à la question précédente. On définit alors $\mathcal{A}_q = (Q \setminus \{q\}, \Sigma, q_0, q_f, \delta')$, où $\delta'(q_1, q_2) = \delta(q_1, q_2) \cup e_{q_1,q_2}$ pour toute paire d'états $q_1, q_2 \in Q \setminus \{q\}$. Montrer que $\mathcal{A}$ et $\mathcal{A}_q$ reconnaissent le même langage.

Exercice 8. Montrer qu'à tout automate on peut associer une expression régulière qui reconnait le même langage.

3 Puzzles

Exercice 9. Pour tout alphabet Σ, donner l'ensemble des mots $(x, y) \in (\Sigma^*)^2$ tels que $xy = yx$.

Exercice 10. Expliciter la forme des langages rationnels unaires (c'est-à-dire, sur un alphabet à une lettre).

4 Rationalité

Exercice 11. Étant donné un langage rationnel L sur un alphabet Σ, prouver que les langages suivants sont rationnels.

a) $Init(L) = \{u \in \Sigma^* \mid \exists v \in \Sigma^*, uv \in L\}$.

b) $Min(L) = \{w \in L \mid \nexists u \in L, u$ préfixe propre de $w\}$.

c) $Max(L) = \{w \in L \mid \forall u \in \Sigma^*, wu \in L \Rightarrow u = \varepsilon\}$.

d) $Cycle(L) = \{uv \in \Sigma^* \mid vu \in L\}$.

e) $\frac{1}{2}L = \{u \in \Sigma^* \mid \exists v \in \Sigma^*, uv \in L \wedge |v| = |u|\}$.

5 Automates déterministes synchronisants

Définition 3. *Un automate déterministe $\mathcal{A} = (\Sigma, Q, q_i, F, \delta)$ est dit* synchronisant *s'il existe un mot w et un état q tels que pour tout $p \in Q$, $\hat{\delta}(p, w) = q$.*

Pour tout $n \geq 3$, on définit l'automate $\mathcal{C}_n = (\llbracket 0, n-1 \rrbracket, \{a, b\}, 0, 1, \delta)$ où $\delta(0, a) = 1$ et $\delta(q, a) = q$ si $q \neq 0$, et $\delta(n-1, b) = 0$ et $\delta(q, b) = q + 1$ sinon. Informellement, la lettre a laisse invariants tous les états sauf l'état 0, et la lettre b effectue une permutation circulaire des états.

Exercice 12. Montrer que le mot $(ab^{n-1})^{n-2}a$, de longueur synchronise $\mathcal{C}_n$.

En fait, on peut même montrer que ce mot est le plus court mot synchronisant pour $\mathcal{C}_n$. Plus généralement, on appelle *délai de synchronisant* la longueur du plus court mot synchronisant.

Conjecture 4 (Černý)**.** *Tout automate synchronisant a un délai de synchronisation d'au plus* $(n-1)^2$.

Cette conjecture est toujours ouverte à ce jour. On montre ici une majoration moins ambitieuse.

Exercice 13. Donner un algorithme permettant de décider si un automate $\mathcal{A} = (Q, \Sigma, q_0, F, \delta)$ est synchronisant. On pourra utiliser l'automate $\hat{\mathcal{A}}$ des parties donné par

$$\hat{\mathcal{A}} = (\mathcal{P}(Q), \Sigma, Q, \{P \subseteq Q \mid |P| = 1\}, \Delta)$$

où $\Delta(P, a) = \{q \in Q \mid \exists p \in P \ \delta(p, a) = q\}$ pour tout $P \subseteq Q$ et $a \in \Sigma$.

Exercice 14. Montrer qu'un automate est synchronisant si et seulement si pour toute paire p, p' d'états, il existe un mot w tel que $\hat{\delta}(p, w) = \hat{\delta}(p', w)$.

Exercice 15. Déduire de la question précédente un algorithme en temps polynomial en le nombre d'états décidant si un automate est synchronisant.

Exercice 16. Soit $\mathcal{A}$ un automate synchronisant à n états et soit P un ensemble de ses états de cardinal k. Montrer qu'il existe un mot w de longueur au plus $1 + \binom{n}{2} - \binom{k}{2}$ tel que

$$\left| \{q \in Q \mid \exists p \in P \ \hat{\delta}(p, w) = q\} \right| < k$$

En déduire que le délai de synchronisation de $\mathcal{A}$ est au plus $(n^3 - n^2)/3$.