Chapitre 2 : Automates finis

Vincent Guisse
vincent.guisse@esisar.grenoble-inp.fr

Grenoble INP - ESISAR

3^{e} année IR & C

14 mars 2022

Idée

On va lire un mot w de gauche à droite pour décider s'il appartient à un langage L, en utilisant une mémoire fixe indépendante de la longueur de w.

Exemple informel

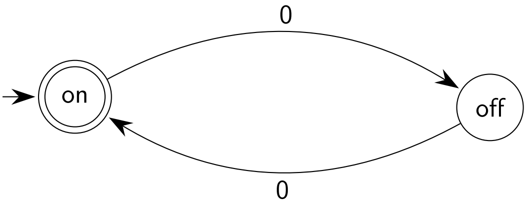
Donnée : Un mot sur $\{0, 1, \ldots, 9\}^*$

Question : Le mot donné code-t-il en base dix un entier divisible par 3 ?

Instances : 291, 507 235.

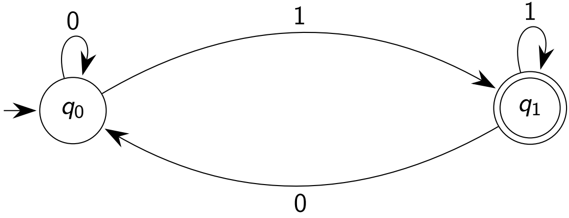
On peut lire successivement les chiffres en vérifiant que le premier n'est pas 0 et en calculant leur somme modulo 3 : 5 états devraient suffire dans la mémoire, indépendamment de la taille du mot.

Automate modélisant les deux états d'un interrupteur à un seul bouton.



$\Sigma = \{0\}$
Sur l'entrée 000, l'automate termine dans l'état Off.
$L = (00)^*$

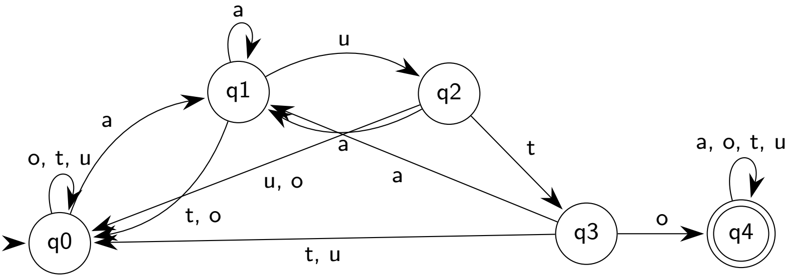
L'automate suivant modélise un interrupteur à deux boutons.



Le "1" correspond à un appui sur le bouton on, le "0" à un appui sur le bouton off.

$L = (0+1)^*1$ des mots sur $\{0, 1\}$ se terminant par 1.

L'automate suivant reconnaît les mots sur $\Sigma = \{a, u, t, o\}$ contenant le mot *auto*. Autrement dit il reconnaît le langage dénoté par l'expression régulière $(a + u + t + o)^* auto(a + u + t + o)^*$.

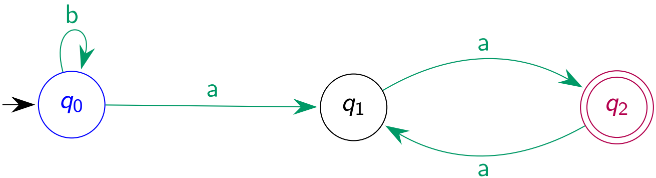


Automate Fini Déterministe

Un Automate Fini déterministe (AFD) est défini par un quintuplet $A = (Q, \Sigma, \delta, q_0, F)$ où :

- Q est un ensemble fini d'états.
- Σ est un alphabet.
- $\delta \ : \ Q \times \Sigma \to Q$ est la fonction de transition.
- $q_i \in Q$ est l'état initial.
- $F \subset Q$ est l'ensemble des états accepteurs.

Lorsque δ est totale on dit que l'AFD est complet.



$\Sigma = \{a, b\}$

Exercice

Faire le diagramme de transitions de l'automate $\mathcal{A} = (Q,\ \Sigma,\ \delta,\ q_i, F)$ avec :

- $Q = \{q_0, q_1, q_2\}$
- $\Sigma = \{a, b\}$
- $q_i = q_0$
- $F = \{q_0, q_1\}$

δ :

q	σ	$\delta(q,\sigma)$
q_0	a	q_1
q_0	b	q_0
q_1	a	q_2
q_1	b	q_0

Est-il complet ? Décrire l'ensemble des mots dont l'exécution termine sur un état accepteur.

Définition : configuration

Une configuration de l'automate $\mathcal{A} = (Q,\ \Sigma,\ \delta,\ q_i, F)$ est un couple $(q, w) \in Q \times \Sigma^*$.

Définition : relation de dérivation

On dit que la configuration $(q, \sigma w)$ se dérive en une étape en la configuration (q', w) lorsque $\sigma \in \Sigma$ et que $\delta(q, \sigma) = q'$. On le note :

$$(q, \sigma w) \vdash_{\mathcal{A}} (q', w)$$

On note $\vdash^*$ la clôture réflexive transitive de la relation $\vdash_{\mathcal{A}}$:

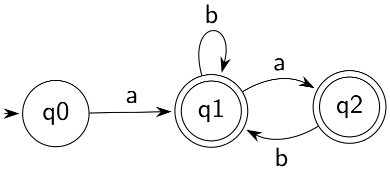
$$\vdash^*_{\mathcal{A}} = \bigcup_{n \geqslant 0} \vdash^n_{\mathcal{A}}$$

Définition : langage accepté

Un mot $w \in \Sigma^*$ est accepté par l'automate $\mathcal{A}$ si et seulement si $(q_i, w) \vdash_{\mathcal{A}}^* (q, \epsilon)$ avec $q \in F$.

Le langage $L(\mathcal{A})$ de l'automate $\mathcal{A}$ est le langage sur Σ des mots acceptés par $\mathcal{A}$:

$$L(\mathcal{A}) = \{w \in \Sigma^* \mid (q_i, w) \vdash_{\mathcal{A}}^* (q, \epsilon) \text{ avec } q \in F\}.$$



$w_1 = abbaba \qquad w_2 = ababaa$

État	ruban
q_0	$\underline{a}bbabab$
q_1	$a\underline{b}baba$
q_1	$ab\underline{b}aba$
q_1	$abb\underline{a}ba$
q_2	$abba\underline{b}a$
q_1	$abbab\underline{a}$
q_2	$abbaba\underline{\ }$

État	ruban
q_0	$\underline{a}babaab$
q_1	$a\underline{b}abaab$
q_1	$ab\underline{a}baab$
q_2	$aba\underline{b}aab$
q_1	$abab\underline{a}ab$
q_2	$ababa\underline{a}b$
	$ababaab$

On va autoriser 3 nouvelles choses :

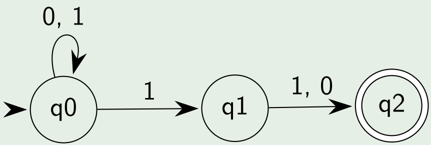
- Des transitions vers plusieurs états
- Les transition sur le mot vide
- Des transition sur des mots (pas d'une seule lettre)

Définition : AFN

Un automate fini non déterministe (AFN) est la donnée d'un quintuplet $(Q, \Sigma, \delta, q_i, F)$ où :

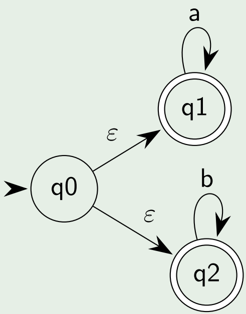
- $Q = \{q_0, q_1, \ldots, q_n\}$ est un ensemble fini d'états.
- Σ est l'alphabet des mots pouvant être lus par l'automate.
- $\delta \subset Q \times \Sigma^* \times Q$ est la relation de transition, c'est une partie finie de $Q \times \Sigma^* \times Q$.
- $q_i \in Q$ est l'état initial.
- $F \subset Q$ est l'ensemble des états accepteurs.

Exemple : un AFN



- Remarquer que de l'état q_0, en lisant le caractère 1 l'automate passe à la fois dans l'état q_0 et dans l'état q_1.
- Avec une définition raisonnable, cet automate accepte le langage des mots dont l'avant dernier caractère est 1 : $(0+1)^*1(0+1)$.

Exemple : un AFN avec des transitions sur le mot vide



Langage accepté : $a^* + b^*$.

Définition : relation de dérivation pour un AFN $\mathcal{A}$

On dit que la configuration $(q, w_0 w)$ se dérive en une étape en la configuration (q', w) lorsque $w_0 \in \Sigma^*$ et que $(q, w_0, q') \in \delta$. On le note :

$$(q, w_0 w) \vdash_{\mathcal{A}} (q', w)$$

On note $\vdash^*$ la clôture réflexive transitive de la relation $\vdash$.
Un mot $w \in \Sigma^*$ est accepté par $\mathcal{A}$ ssi $(q_i, w) \vdash^*_{\mathcal{A}} (q, \epsilon)$ avec $q \in F$.
Le langage $L(\mathcal{A})$ de l'automate $\mathcal{A}$ est le langage sur Σ des mots acceptés par $\mathcal{A}$:

$$L(\mathcal{A}) = \{w \in \Sigma^* \mid (q_i, w) \vdash^*_{\mathcal{A}} (q, \epsilon) \text{ avec } q \in F\}.$$

La différence essentielle avec les automates déterministes est que plusieurs dérivations sont possibles à partir d'une même configuration. Un mot est accepté ssi l'une d'entre elles termine sur un état accepteur.

Exercice

On considère les langages suivants définis sur l'alphabet $\Sigma = \{0, 1\}$

1. $L_1 = \{w \in \Sigma^*, w$ commence par 1 et se termine par $0\}$
2. $L_2 = \{w \in \Sigma^*, w$ comporte trois 1 consécutifs et se termine par $0\}$
3. $L_3 = \{w \in \Sigma^*, w$ commence par 1 et est de longueur paire$\}$

Déterminer des automates finis reconnaissant ces langages.

Définition : automates équivalents

Deux automates sont équivalents lorsqu'ils acceptent le même langage.

Théorème

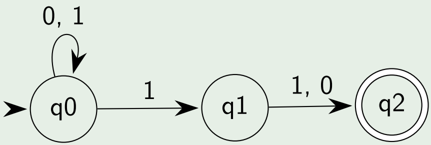
Tout automate fini non déterministe est équivalent à un automate fini déterministe.

Ainsi on n'a pas augmenté les capacités de calcul (=résolution de problèmes de décisions = reconnaissance de langage) de notre modèle.

Avant de prouver le théorème, regardons quelques exemples.

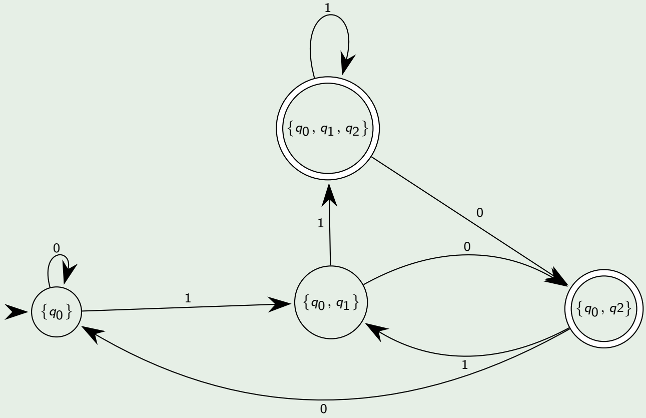
Exercice : déterminisation

Donner un AFD équivalent à cet AFN :

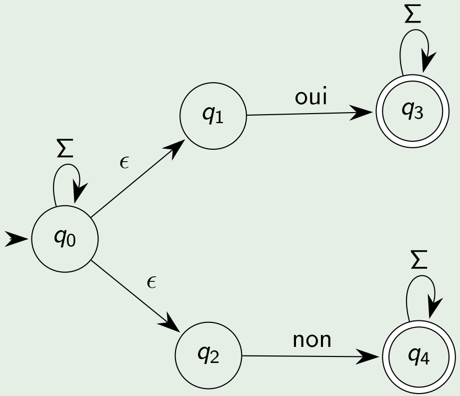


Indication : comme ensemble d'état, prendre $\mathcal{P}(\{q_0, q_1, q_2\})$

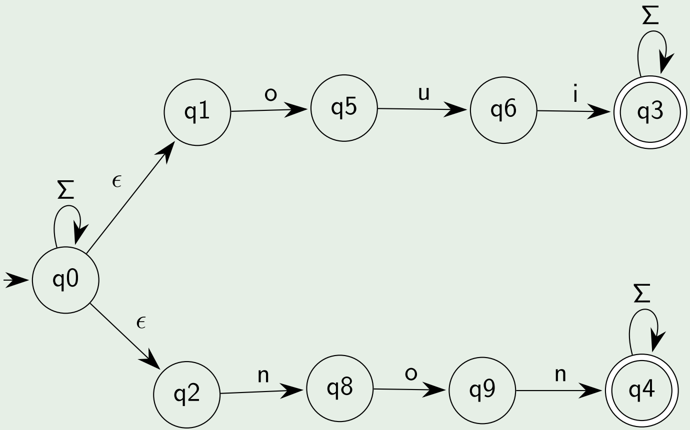
Exercice : réponse



Exemple : élimination des transitions sur les mots de longueur > 1



Exemple : élimination des transitions sur les mots de longueur > 1



Définition : ε-clôture d'un état

Soit $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ un AFN. Alors pour tout état $q \in Q$ on définit l'$\varepsilon-$ clôture $E(q)$ de q comme étant l'ensemble des états accessibles depuis q par une succession de transitions sur le mot vide :

$$E(q) = \{q' \in Q \,|\, (q, \varepsilon) \vdash^* (q', \varepsilon)\}$$

Exemple :

Dans l'exemple précédent, $E(q_0) = \{q_0, q_1, q_2\}$, $E(q_1) = \{q_1\}$.

Soit $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ un AFN.
Construisons un AFD $\mathcal{A}' = (Q', \Sigma, q_0', \delta', F')$ tel que $L(\mathcal{A}) = L(\mathcal{A}')$.
Quitte à rajouter des états, on suppose que $\mathcal{A}$ ne comporte de transitions que sur des mots de longueur $\leqslant 1$.
On prend :

- $Q' = \mathcal{P}(Q)$,
- $q_0' = E(q_0)$,
- $F' = \{q' \in Q' \mid q' \cap F \neq \emptyset\}$, l'ensemble des parties de Q qui comportent au moins un état accepteur.

Il reste à construire δ' :

$$\begin{array}{rll} \delta' : & \mathcal{P}(Q) \times \Sigma & \longrightarrow \mathcal{P}(Q) \\ & (q', \sigma) & \mapsto \delta'(q', \sigma) = \displaystyle\bigcup_{q \in q'} \bigcup_{(q,\sigma,r) \in \delta} E(r) \end{array}$$

Preuve, 2/2 : correction de la construction

On montre par récurrence sur la longueur n d'un mot w de Σ que l'ensemble q' des états de $\mathcal{A}$ atteignables par dérivation complète :

$$q' = \{q \in Q \,|\, (q_0, w) \vdash_{\mathcal{A}}^{*} (q, \varepsilon)\}$$

est vide ou est l'unique état de $\mathcal{A}'$ tel que $(q'_0, w) \vdash_{\mathcal{A}'}^{n} (q', \varepsilon))$:

- Si $|w| = 0$, alors $w = \varepsilon$ et $q' = E(q_1)$, et on a bien $q'_i = E(q_i)$.
- Soit $w = x\sigma$ de longueur $n+1$ avec $\sigma \in \Sigma$. Alors par hypothèse de récurrence, soit q' est vide et la lecture de $x\sigma$ ne donne rien de plus soit :

  $$(q'_0, x) \vdash_{\mathcal{A}'}^{n} (q', \varepsilon)$$

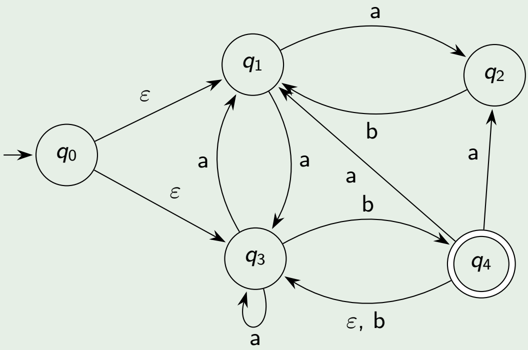
  avec $q' = \{q \in Q \,|\, (q_i, x) \vdash_{\mathcal{A}}^{*} (q, \epsilon)\}$, donc

  $$(q'_i, x\sigma) \vdash_{\mathcal{A}'}^{n} (q', \sigma) \vdash_{\mathcal{A}'} (q'', \epsilon),$$

  où par construction de $\mathcal{A}'$, $q'' = \bigcup_{q \in q'} \bigcup_{(q,\sigma,r) \in \delta} E(r)$ est bien l'ensemble des états de $\mathcal{A}$ atteignables par lecture complète de $w = x\sigma$.

Exercice

Déterminiser l'automate suivant :



Théorème

Un langage est régulier si et seulement si il est reconnu par un AFD.

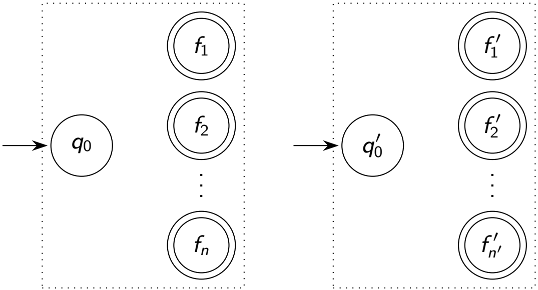
Pour prouver cette équivalence, on va montrer successivement les deux implications :

- $\Rightarrow$ Si un langage est régulier, alors il est reconnu par un AFD.
- $\Leftarrow$ Si un langage est reconnu par un AFD, alors il est régulier.

Régulier $\Rightarrow$ Reconnu par un AFD

Les langages réguliers étant définit par induction structurelle, on fait une preuve par induction :

- Cas de base : On a des automates finis reconnaissant $\emptyset$, $\{\epsilon\}$,et $\{\sigma\}$ pour $\sigma \in \Sigma$.
- Induction : Soit $\mathcal{A}$ et $\mathcal{A}'$ deux AFN avec des états initiaux et accepteurs comme sur le schéma :



- Cas du produit : automate reconnaissant $\mathcal{L}(\mathcal{A})\mathcal{L}(\mathcal{A}')$.
- Cas de la réunion : automate reconnaissant $\mathcal{L}(\mathcal{A}) \cup \mathcal{L}(\mathcal{A}')$.
- Cas de l'étoile : automate reconnaissant $\mathcal{L}(\mathcal{A})^*$.

Régulier $\Leftarrow$ Reconnu par un AFD

Soit $\mathcal{A} = (\{q_1, q_2, \ldots, q_n\}, \Sigma, \delta, q_1, F)$ un AFD, et pour $(i, j, k) \in [\![1, n]\!]^2 \times [\![0, n]\!]$ soit $L(i, j, k)$ l'ensemble des mots permettant de passer de l'état q_i à l'état q_j en passant seulement par des états intermédiaires dans $\{q_1, q_2, \ldots, q_k\}$.

- Pour $k = 0$, il n'y a pas d'état intermédiaire possible, donc $L(i, j, 0)$ est vide ou ne contient que des mots de longueur inférieure à 1, il est régulier.
- Soit $k < n$, supposons que pour tous $(i, j, k) \in [\![1, n]\!]^2 \times [\![0, k]\!]$, $L(i, j, k)$ est régulier. Alors :

$$L(i, j, k+1) = L(i, j, k) + L(i, k+1, k)(L(k+1, k+1, k))^* L(k+1, j, k)$$

  est régulier.

Donc $\mathcal{L}(\mathcal{A}) = \bigcup_{q_j \in F} L(1, j, n)$ est une réunion finie de langages réguliers, il est régulier.

On va explorer une limite facile des langages régulier : un langage régulier étant accepté par un AFN, ses mots sont reconnus par une procédure effective avec une **mémoire finie**. L'idée est que si on reconnaît un mot plus grand que le nombre d'états, alors la dérivation comporte une boucle, et on est obligé d'accepter beaucoup de mots plus grand obtenus en parcourant plusieurs fois cette boucle.

Théorème : lemme de pompage

Soit L un langage régulier et $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ un AFN reconnaissant L. Alors si $w \in L$ avec $|w| \geqslant \text{Card}(Q)$, on a $w = xuy$ avec :

- $|xu| \leqslant \text{Card}(Q)$ et $|u| \geqslant 1$
- Pour tout $n \in \mathbb{N}$, $xu^n y \in L$.

Lemme de pompage : preuve

Soit $w = w_1 w_2 \dots w_n w'$ un mot de longueur supérieur ou égale à n de L, alors comme w est accepté la dérivation :

$$(q_0, w) \vdash_{\mathcal{A}} (q_1, w_2 \dots w_n x) \vdash_{\mathcal{A}} \dots (q_n, x)$$

existe et comporte n+1 états dans Q, donc deux d'entre eux sont égaux, soit q_i et q_j avec $0 \leqslant i < j \leqslant n$, ainsi $u = w_1 \dots w_i$ (mot vide si $i = 0$), $u = w_{i+1} \dots w_j$ et $y = w_{j+1} \dots w_n$ (mot vide si $j = n$) conviennent.

Application : prouver qu'un langage n'est pas régulier.

Idée : on prouve que pour n arbitrairement grand, il existe des mots $w \in L$ avec $|w| \geqslant n$ et tels que pour toute factorisation $w = xuv$, il y ait $p \in \mathbb{N}$ tel que $xu^p y \notin L$.

Exercice

Montrer que $\mathcal{L} = \{a^n b^n \text{ où } n \in \mathbb{N}\}$ n'est pas un langage régulier.

Exercice

Indiquez si les langages suivants sont réguliers. Dans les cas où ils le sont, vous déterminerez des automates finis les reconnaissants.

1. $L_1 = \{a^n b^m,\ (n, m) \in \mathbb{N}, (\exists k \in \mathbb{N}), n + m = 2k + 1\}$
2. $L_2 = \{a^p b^q a^r b^s,\ (p, q, r, s) \in \mathbb{N} \text{ et } p + r = q + s\}$