Automates (1)

option informatique

Les automates finis, c'est l'infini mis à la portée des informaticiens.

Jacques Sakarovitch

Introduction

- Un **automate** est une **machine abstraite** qui peut prendre un nombre n_e fini d'**états**.
- L'état global de l'automate change selon des sollicitations extérieures.

Introduction

- Soit une machine à café qui accepte des pièces de 10, 20 et 50 centimes.
- Un café coûte 40 centimes.
- La machine délivre le café dès que la somme introduite est supérieure ou égale à 40 centimes.
- Elle ne rend pas la monnaie !
- La situation finale est celle pour laquelle 40 centimes au moins ont été introduits dans la machine.

Introduction

L'automate peut être décrit à l'aide d'un **tableau**.

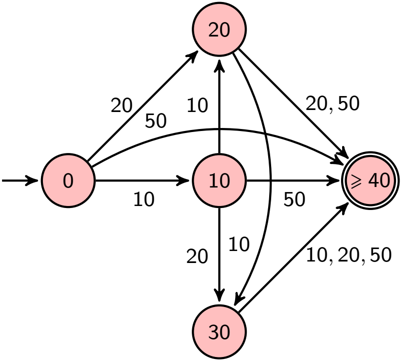
- Les lignes correspondent aux **états** (valeur numéraire possible).
- Les colonnes correspondent aux **sollicitations extérieures** (pièces).
- Aux intersections se trouvent les **états de transition**.

	10	20	50
0	10	20	$\geqslant 40$
10	20	30	$\geqslant 40$
20	30	$\geqslant 40$	$\geqslant 40$
30	$\geqslant 40$	$\geqslant 40$	$\geqslant 40$
$\geqslant 40$			

Introduction

L'automate peut aussi être décrit par un **graphe**.

- Les sommets du graphe sont appelés les **états** de l'automate. Dans l'exemple, c'est la valeur numéraire possible.
- Les arêtes du graphe sont appelées les **transitions étiquetées** par les données. Dans l'exemple, ce sont les valeurs des pièces.



Automate fini déterministe

Automate fini déterministe

Définition 1 (automate fini déterministe)

Un **automate fini déterministe** $\mathcal{A}$ est un quintuplet $(\Sigma, Q, I, F, \delta)$.

- Σ est un **alphabet fini**.
- Q est un ensemble fini dont les éléments sont les **états** de $\mathcal{A}$.
- $I = \{q_0\} \subset Q$ est un singleton où q_0 est l'**état initial** de $\mathcal{A}$.
- $F \subset Q$ est l'ensemble des **états acceptants** de $\mathcal{A}$.
- δ est la **fonction de transition** d'une partie de $Q \times \Sigma$ dans Q.

Automate fini déterministe - Exemple

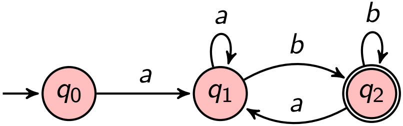
Ensembles de l'automate

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

Fonction de transition

$$\delta(q_0, a) = q_1$$
$$\delta(q_1, a) = q_1 \qquad \delta(q_2, a) = q_1$$
$$\delta(q_1, b) = q_2 \qquad \delta(q_2, b) = q_2$$

Graphe de l'automate



L'état initial est signalé par une flèche entrante, l'état final par un double cercle ou par une flèche sortante.

Automate fini déterministe - Exemple

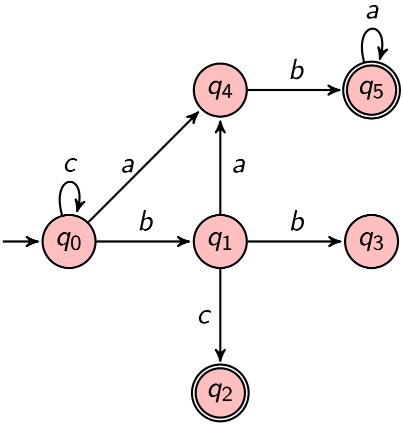
Ensembles de l'automate

$$\Sigma = \{a, b, c\}$$
$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$$
$$I = \{q_0\}$$
$$F = \{q_2, q_5\}$$

Fonction de transition

$$\delta(q_0, a) = q_4 \qquad \delta(q_1, a) = q_4$$
$$\delta(q_0, b) = q_1 \qquad \delta(q_1, b) = q_3$$
$$\delta(q_0, c) = q_0 \qquad \delta(q_1, c) = q_2$$
$$\delta(q_4, b) = q_5 \qquad \delta(q_5, a) = q_5$$

Graphe de l'automate



Automate fini déterministe

Le caractère **déterministe** de l'automate est lié :

- d'une part à l'existence d'un **unique état initial** ;
- d'autre part au fait que δ soit une **fonction**.

Si δ est définie sur $Q \times \Sigma$ tout entier, l'automate $\mathcal{A}$ est dit **complet**.

Reconnaissance d'un langage

- **Lire un automate**, c'est partir d'un état initial et aboutir à un état acceptant en passant par un certain nombre d'états intermédiaires. Le résultat obtenu est un mot de Σ^*.
- Étant donnés un automate $\mathcal{A}$ et un mot $u \in \Sigma^*$, la question se pose de savoir si u est reconnu par $\mathcal{A}$. Plus généralement, étant donné un automate $\mathcal{A}$ et un langage L, si tous les mots de L sont reconnus par $\mathcal{A}$, on dit que l'**automate reconnaît le langage** L.

Exemple

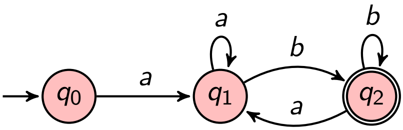
Ensembles de l'automate

$$\Sigma = \{a, b\} \quad Q = \{q_0, q_1, q_2\}$$
$$I = \{q_0\} \quad F = \{q_2\}$$

Fonction de transition

$$\delta(q_0, a) = q_1$$
$$\delta(q_1, a) = q_1 \quad \delta(q_2, a) = q_1$$
$$\delta(q_1, b) = q_2 \quad \delta(q_2, b) = q_2$$

Graphe de l'automate

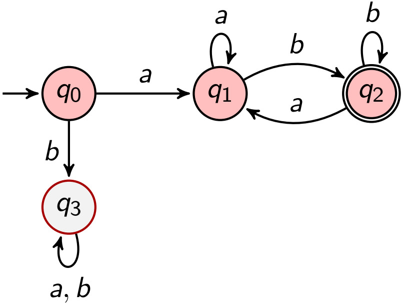


- Cet **automate** lit les mots qui débutent par un a et se terminent par un b. Il **reconnaît** le **langage** $a\Sigma^* b$.
- Mais ne peut pas lire les mots qui commencent par un b car la transition $\delta(q_0, b)$ n'existe pas. On dit que le couple (q_0, b) est **bloquant**.

☛ **Pour s'entraîner :** exercice 1 (td automates (1))

Automate complet

- Un **automate complet** est un automate **sans blocage**.
- Pour rendre un automate complet, on peut ajouter un état inutile appelé **puits**.



Automate avec puits q_3

Chemin

- Une **transition dans un automate $\mathcal{A}$** associe un état q_j à un couple (q_i, a). Si δ est la fonction de transition, on écrit :

$$\delta(q_i, a) = q_j \qquad \text{ou} \qquad q_i \xrightarrow{a} q_j$$

- Un **chemin dans un automate $\mathcal{A}$** est une suite finie de transitions consécutives.

$$q_{deb} \xrightarrow{a_1} q_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} q_{fin}$$

  Le mot $a_1 \dots a_n$ est appelé **étiquette du chemin**. q_{deb} est l'état initial du chemin et q_{fin} son état d'arrivée.

- Un chemin est **acceptant** quand son état d'arrivée est un état acceptant d'un automate. Un **mot** de Σ^* est **reconnu par un automate** $\mathcal{A}$ s'il est l'étiquette d'un chemin acceptant. Le langage $L(\mathcal{A})$ reconnu par $\mathcal{A}$ est l'ensemble des mots reconnus par $\mathcal{A}$.

☛ **Pour s'entraîner :** exercice 7 (td automates (1))

États accessible, co-accessible, utile

- Un état q est **accessible** s'il existe un chemin menant de l'état initial q_0 à q.
- Un état q est **co-accessible** s'il existe un chemin menant de q à un état acceptant.
- Un état q est **utile** s'il est est accessible et co-accessible.

Automate émondé

- Un **automate émondé** est un automate qui ne possède que des états utiles. *(émonder = débarrasser du superflu)*
- Pour obtenir un automate émondé à partir d'un automate, on supprime les états qui ne sont pas utiles et leurs transitions.
- Deux **automates** sont **équivalents** lorsqu'ils reconnaissent le même langage.

☛ **Pour s'entraîner :** exercices 3, 4 (td automates (1))

Automate émondé - Intérêt

- La reconnaissance d'un mot par un automate se fait en parcourant successivement chacun de ses caractères.
- Un automate complet permet la reconnaissance de mots avec une complexité algorithmique proportionnelle à la longueur du mot.
- Mais il arrive que la lecture d'un mot aboutisse à un état bloquant (puits), impossible à quitter. La lecture du mot doit alors s'arrêter.
- Par ailleurs, l'état bloquant n'étant pas acceptant, il peut être supprimé sans modifier le langage reconnu par l'automate.
- L'automate équivalent ainsi obtenu est l'**automate émondé**.

☛ **Pour s'entraîner :** exercices 5, 6 (td automates (1))

Automate fini non déterministe

Automate non déterministe

Un **automate déterministe** a un **unique état initial** et la **relation de transition** est une **fonction** : à chaque état est associée, au plus, une transition d'étiquette donnée.

Définition 2 (automate non déterministe)

Un **automatique fini non déterministe** $\mathcal{A}$ est un quintuplet $(\Sigma, Q, I, F, \delta)$ où :

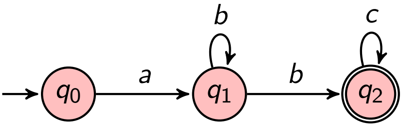
- Σ est un alphabet fini ;
- Q est l'ensemble fini des **états** de $\mathcal{A}$;
- $I \subset Q$ est l'ensemble des **états initiaux** ;
- $F \subset Q$ est l'ensemble des **états acceptants** ;
- δ est une application d'une partie d'une partie de $Q \times \Sigma$ dans $\mathcal{P}(Q)$ appelée **fonction de transition**.

Dans un tel automate, la notion de blocage n'a plus de sens puisque $\delta(q, a)$ peut prendre la valeur $\varnothing$.

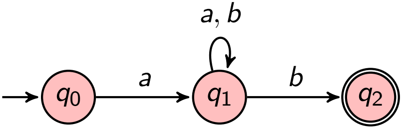
Automate non déterministe

Les automates suivants ne sont pas déterministes.

- (q_1, b) admet deux images par la fonction de transition δ.

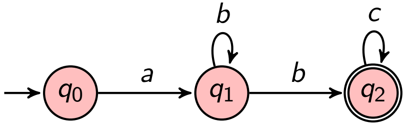


- (q_1, b) admet deux images par la fonction de transition δ.

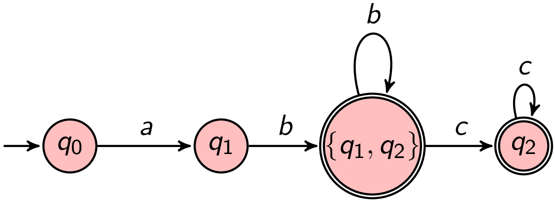


Automate non déterministe

- Soit l'automate non déterministe suivant.



- On peut rendre un automate déterministe en regroupant toutes les images d'un couple donné en un seul état. En regroupant q_1 et q_2, on peut définir un nouvel état $\{q_1, q_2\}$ et une nouvelle relation de transition δ' qui est une fonction.



Automate non déterministe

- Lors de la recherche d'un automate susceptible de reconnaître un langage, les automates non déterministes sont souvent plus faciles à construire que les automates déterministes.
- Le nombre d'états d'un automate non déterministe peut être notablement inférieur à celui de l'automate déterministe équivalent.
- Un automate non déterministe est moins efficace que son équivalent déterministe lors de la reconnaissance d'un mot. Avec un automate déterministe, un mot est déterminé de manière unique par une étiquette unique.

Théorème 3 (résultat fondamental)

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

Automate non déterministe

- Une **transition** est un triplet (q_i, a, q_j) tel que $q_j \in \delta(q_i, a)$. On la note aussi $q_i \xrightarrow{a} q_j$.
- Un **chemin dans un automate $\mathcal{A}$** est un suite finie de transition consécutive.

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} q_n$$

- Le mot $a_1 \dots a_n$ désigne l'**étiquette du chemin**.

Déterminisation - Construction par sous-ensembles

- Soit $\mathcal{A} = (\Sigma, Q, I, F, \delta)$ un **automate non déterministe** à n_e états. On pose :

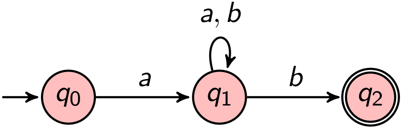
$$F' = \{P \in \mathcal{P}(Q) | P \cap F \neq \emptyset\}$$

$$\forall (P, a) \in \mathcal{P}(Q) \times \Sigma \qquad \delta'(P, a) = \bigcup_{q \in P} \delta(q, a)$$

- $\mathcal{A}' = (\Sigma, \mathcal{P}(Q), I, F', \delta')$ est un **automate déterministe** à 2^{n_e} états, équivalent à l'automate $\mathcal{A}$.

Déterminisation - Exemple

- Automate non déterministe $\mathcal{A} = (\Sigma, Q, I, F, \delta)$



- Ensembles Σ, Q, I, F

$$\Sigma = \{a, b\} \qquad Q = \{q_0, q_1, q_2\} \qquad I = \{q_0\} \qquad F = \{q_2\}$$

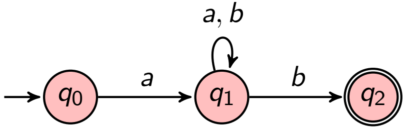
- Relation de transition δ

δ	a	b
q_0	q_1	$-$
q_1	q_1	$\{q_1, q_2\}$
q_2	$-$	$-$

Déterminisation - Exemple

- Parties de Q

$$\{\varnothing, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0, q_1\}, \{q_1, q_2\}, \{q_0, q_2\}, \{q_0, q_1, q_2\}\}$$



- Relation de transition δ' (ébauche)

δ'	a	b	δ'	a	b
$\varnothing$	$\varnothing$	$\varnothing$	$\{q_0, q_1\}$	$\{q_1\}$	$\{q_1, q_2\}$
$\{q_0\}$	$\{q_1\}$	$\varnothing$	$\{q_1, q_2\}$	$\{q_1\}$	$\{q_1, q_2\}$
$\{q_1\}$	$\{q_1\}$	$\{q_1, q_2\}$	$\{q_0, q_2\}$	$\{q_1\}$	$\varnothing$
$\{q_2\}$	$\varnothing$	$\varnothing$	$\{q_0, q_1, q_2\}$	$\{q_1\}$	$\{q_1, q_2\}$

Déterminisation - Exemple

- Relation de transition δ' (ébauche)

δ'	a	b	δ'	a	b
$\varnothing$	$\varnothing$	$\varnothing$	$\{q_0, q_1\}$	$\{q_1\}$	$\{q_1, q_2\}$
$\{q_0\}$	$\{q_1\}$	$\varnothing$	$\{q_1, q_2\}$	$\{q_1\}$	$\{q_1, q_2\}$
$\{q_1\}$	$\{q_1\}$	$\{q_1, q_2\}$	$\{q_0, q_2\}$	$\{q_1\}$	$\varnothing$
$\{q_2\}$	$\varnothing$	$\varnothing$	$\{q_0, q_1, q_2\}$	$\{q_1\}$	$\{q_1, q_2\}$

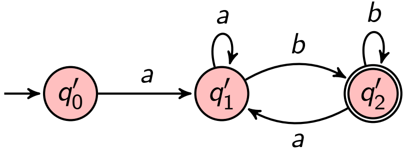
- Suppression des états non accessibles et de l'état vide.
  Relation de transition δ' (version améliorée)

δ'	a	b
$\{q_0\}$	$\{q_1\}$	$-$
$\{q_1\}$	$\{q_1\}$	$\{q_1, q_2\}$
$\{q_1, q_2\}$	$\{q_1\}$	$\{q_1, q_2\}$

- Nouveaux états : $q'_0 = \{q_0\}$ $\quad q'_1 = \{q_1\}$ $\quad q'_2 = \{q_1, q_2\}$.

Déterminisation - Exemple

- Automate déterministe $\mathcal{A}' = (\Sigma, Q', I', F', \delta')$ équivalent à $\mathcal{A}$



- Ensembles A', Q', I', F'

$$\Sigma = \{a, b\} \qquad Q' = \{q'_0, q'_1, q'_2\} \qquad I' = \{q'_0\} \qquad F' = \{q'_2\}$$

- Relation de transition δ' (version finale)

δ'	a	b
$\{q'_0\}$	$\{q'_1\}$	–
$\{q'_1\}$	$\{q'_1\}$	$\{q'_2\}$
$\{q'_2\}$	$\{q'_1\}$	$\{q'_2\}$