Théorie des Langages Automates

Yannick Le Nir Gaspard Férey

Contributions de :
Taisa Guidini Goncalves

CY TECH

yannick.lenir@eisti.fr gaspard.ferey@inria.fr

Automates

Concepts

- Un automate est une procédure effective (un algorithme) permettant de déterminer si un mot donné appartient à un langage.
- Notion d'état : description du système à un instant donné.
- Notion d'évènement : communication avec l'extérieur.
- Notion de transitions d'états : modélisent l'évolution du système $e_i \xrightarrow{evt} e_f$

Yannick Le Nir, Gaspard Férey

Contributions de : Taisa Guidini Goncalves

Exemple d'automate

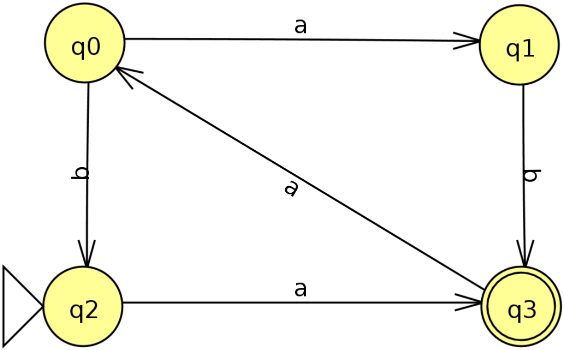


Figure – Automate

Etats : $\{q0, q1, q2, q3\}$ Evènements : $\{a, b\}$ Transitions :
$\{q2 \xrightarrow{a} q3,\ q3 \xrightarrow{a} q0,\ q0 \xrightarrow{a} q1,\ q0 \xrightarrow{b} q2,\ q1 \xrightarrow{b} q3\}$

Automate fini

Définition

Un automate fini est un quintuplet $< Et, F, Ev, \Psi, q_0 >$ tel que :

- Et est un ensemble fini, non vide d'états
- Ev est un ensemble fini, non vide d'évènements
- $\Psi \subset Et \times Ev \times Et$ est une relation de transitions
- $F \subset Et$ est un ensemble d'états finaux
- $q_0 \in Et$ est l'état initial

Transitions

Modélisation

Les transitions entre états peuvent modéliser des problèmes autres que liés à l'axe du temps

Non-évènement

Une transition associée à un non-évènement, noté ε est appelée ε-transition (exemple : attente d'un message qui n'arrive pas).

Déterminisme

Définition

- Un automate est dit déterministe si pour tout état $et_i \in Et$ et pour tout évènement $ev \in Ev$, il existe au plus un état $et_f \in Et$ tel que $(et_i, ev, et_f) \in \Psi$.
- C'est-à-dire qu'il a un seul état initial et que, pour chaque "état de départ" il existe une seule transition possible avec le même évènement.
- Un automate qui n'est pas déterministe est dit non déterministe.

Représentation

Si $< Et, F, Ev, \Psi, q_0 >$ est déterministe, on peut choisir de modéliser les transitions par une fonction partielle

$$\Psi : Et \times Ev \to Et$$

et de la représenter par une table de transitions.

Exemple d'automate fini non déterministe

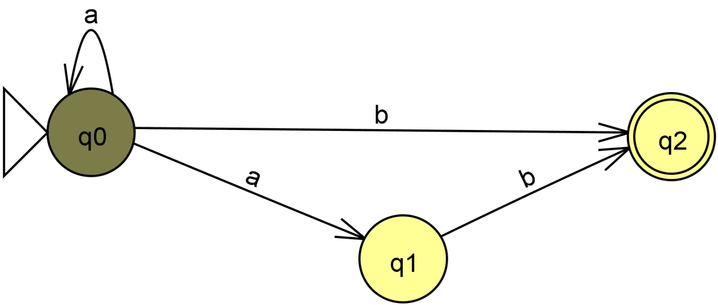


Figure – Automate fini non déterministe

Langage

Ensemble de motifs

Soit G un ensemble et C une condition portant sur les éléments de G. On définit l'ensemble de motifs de G associés à C par :

$$M_C(G) = \{g \in G / C(g) \text{ est vraie}\}$$

Définition d'un langage

Comment définir un langage sur un alphabet A ?

- Définition ensembliste $\quad \{a^n b^n\}$
- Par l'ensemble des motifs associés à un condition sur les mots de A $\quad \{w \in A^* / |w| = 6\}$
- Par une grammaire $G =< T, N, S, P >$ dont les symboles terminaux sont dans A $\quad (T \subset A)$

Langage

Ensemble de motifs

Soit G un ensemble et C une condition portant sur les éléments de G. On définit l'ensemble de motifs de G associés à C par :

$$M_C(G) = \{g \in G / C(g) \text{ est vraie}\}$$

Définition d'un langage

Comment définir un langage sur un alphabet A ?

- Définition ensembliste $\quad \{a^n b^n\}$
- Par l'ensemble des motifs associés à un condition sur les mots de A $\quad \{w \in A^* / |w| = 6\}$
- Par une grammaire $G =< T, N, S, P >$ dont les symboles terminaux sont dans A $\quad (T \subset A)$
- Par un automate !

Langage reconnaissable

Soit $A =< Et, F, Ev, \Psi, q_0 >$ un automate fini.

Definition

Le langage $L(A)$ reconnu par A est l'ensemble des mots $u_1 \cdot u_2 \cdot \ldots \cdot u_n$ tels qu'il existe au moins une configuration où l'automate après avoir été initialisé à l'état initial q_0 puis soumis successivement aux événements $u_1,\ u_2,\ \ldots,\ u_n$ se trouve dans un état terminal de F.

$$u_1 \cdot u_2 \cdot \ldots \cdot u_n \in L(A) \quad \Leftrightarrow \quad q_0 \xrightarrow{u_1} q_1 \xrightarrow{u_2} \ldots \xrightarrow{u_n} q_n \in F$$

Remarque

L'alphabet du langage $L(A)$ est inclut dans Ev : $L(A) \subset Ev^*$

Exemple d'automate

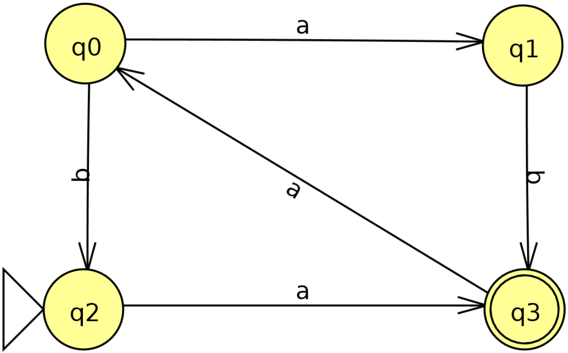


Figure – Automate fini déterministe

q2

q_0 = état initial $\quad F = \{ \text{q3} \}$

Chaînes reconnues : a, aaba, aaab, ...

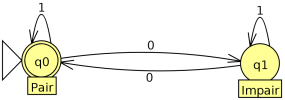
Exemple d'automate (2)

Exemple

Soit G l'ensemble des suites finies de nombres binaires (0 ou 1) et C la condition définie par $C(g)$: g contient un nombre pair de 0. L'automate fini suivant reconnaît les nombres respectant la condition C :

- $Et = \{\text{Pair}, \text{Impair}\}$
- $F = \{\text{Pair}\}$
- $Ev = \{0, 1\}$
- Ψ est définie par les transitions

Evènement Etat	0	1
Pair	Impair	Pair
Impair	Pair	Impair



Déterminisation

Théorème

Pour tout automate A, il existe un automate déterministe A' reconaissant le même langage : $L(A) = L(A')$.

Preuve

On définira plus tard un algorithme dit de déterminisation permettant de construire A' à partir de A.

Autres propriétés

Soit A un automate fini.

- Il existe A' complet tel que $L(A') = L(A)$.
- Il existe A' sans ε-transitions tel que $L(A') = L(A)$.

Rappel

Définition

Une grammaire $G =< T, N, S, P >$ est de type 3 si les règles de P sont toutes de la forme

- $X \to a$
- $X \to bY$
- $X \to \varepsilon$

avec $X, Y \in N$, $a, b \in T$.
Une telle grammaire est aussi dite régulière ou rationnelle.

Définition

Un langage est de type 3 (on dit aussi régulier ou rationnel) s'il peut être engendré par une grammaire de type 3.

Langages réguliers et AFD

Théorème de Kleene

Tout langage reconnu par un automate fini est de type 3.

Théorème de Kleene

Tout langage de type 3 est reconnu par un automate fini.

Corollaire

Les grammaires de type 3 et les automates finis déterministes permettent de définir la même classe de langages : les langages réguliers.

Construction d'un automate fini à partir d'une grammaire de type 3

Soit $G =< T, N, S, P >$ une grammaire de type 3.
On construit l'automate $A =< Et, F, Ev, \Psi, q_0 >$ par :

- Choisir $Ev = T$
- Choisir $Et = N \cup \{\mathbf{T}\}$ avec $\mathbf{T} \in F$.
- Choisir $q_0 = S$
- Pour toute règle $X \to zY \in P$:
  Générer une transition $X \xrightarrow{z} Y \in \Psi$.
- Pour toute règle $X \to \varepsilon \in P$:
  Définir X comme état final : $X \in F$
- Pour toute règle $X \to z \in P$:
  Générer une transition $X \xrightarrow{z} \mathbf{T} \in \Psi$.

Automates et grammaires

De la même façon, on peut également construire une grammaire générant exactement le langage reconnu par un automate fini A.

Remarque

L'automate généré à partir d'une grammaire G est en général non déterministe cependant d'après le théorème de déterminisation, il existe également un automate déterministe reconnaissant le langage généré par G.
On peut même en trouver un complet et sans ε-transition.

Exemple d'application

Analyse lexicale de l'affectation numérique de type 3

Une grammaire $G =< T, N, mot, P >$ peut être définie par :

- $T = lettre \cup chiffre \cup \{+, -, *, /, =, .\}$
- $N = \{mot, chaine, entier, reel, point\}$
- $P = \left\{ \begin{array}{ll} mot \rightarrow + | - | * | / & chaine \rightarrow lettre\ chaine \\ mot \rightarrow = & chaine \rightarrow chiffre\ chaine \\ mot \rightarrow chiffre\ entier & chaine \rightarrow \varepsilon \\ mot \rightarrow lettre\ chaine & \\ entier \rightarrow chiffre\ entier \mid \varepsilon & point \rightarrow chiffre\ reel \\ entier \rightarrow .\ point & reel \rightarrow chiffre\ reel \mid \varepsilon \end{array} \right\}$

Langage reconnaissable par l'automate suivant :

	a..z A..Z lettre	0..9 chiffre	+ - / *	=	.
mot	chaine	entier	operateur	affectation	
chaine	chaine	chaine			
entier		entier			point
point		réel			
réel		réel			
opérateur					
affectation					

Analyse lexicale pour l'affectation numérique

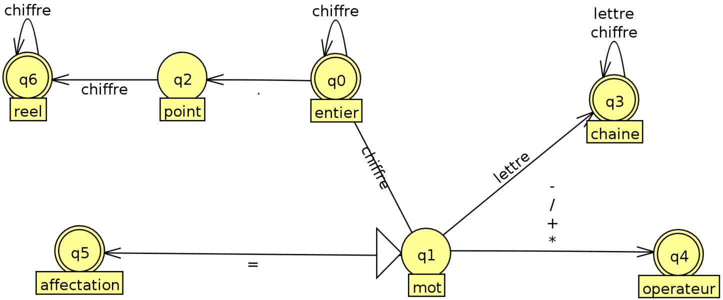


Figure – Automate fini déterministe