INF 302 : Langages & Automates

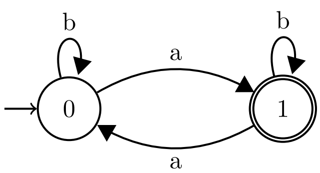
Chapitre 3 : Automates déterministes

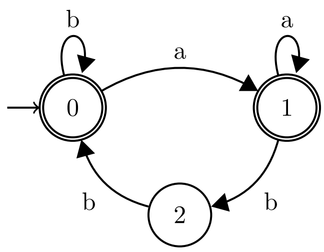
Yliès Falcone

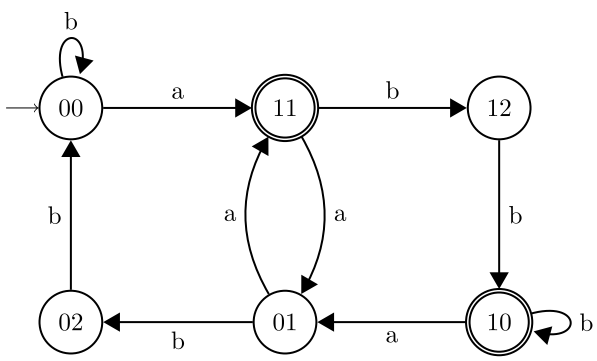
ylies.falcone@univ-grenoble-alpes.fr — www.ylies.fr

Univ. Grenoble-Alpes
-
Laboratoire d'Informatique de Grenoble - www.liglab.fr

Intuition et objectifs







- Ingrédients de base : états (accepteurs), symboles, transitions — *syntaxe*.
- Exécution, mot accepté, langage accepté — *sémantique*.

Plan Chap. 3 - Automates déterministes

Univ. Grenoble Alpes, Département Licence Sciences et Technologies, Licence deuxième année

Plan Chap. 3 - Automates déterministes

À propos des automates à états finis déterministes

Définition formelle des automates à états finis déterministes.

Dans les automates à états finis déterministes :

- *déterministe* réfère au fait que pour un mot d'entrée, l'automate est dans un état seul état à la fois ;
- *fini* réfère au fait que l'automate à un nombre fini d'états.

Dans la suite, nous dirons *automate déterministe*.

Dans les prochains cours, nous étudierons les automates *non-déterministes*.

Univ. Grenoble Alpes, Département Licence Sciences et Technologies, Licence deuxième année

Automate déterministe

◁ Ingrédients de la définition d'un automate

Définition (Automate déterministe)

Un automate déterministe (abrégé AD) est donné par un 5-tuple

$$(Q, \Sigma, q_{\mathrm{init}}, \delta, F)$$

tel que :

- Q est un ensemble non-vide dont les éléments sont appelés états ;
- Σ est l'alphabet de l'automate ;
- $q_{\mathrm{init}} \in Q$ est l'état initial ;
- $\delta : Q \times \Sigma \to Q$ est la fonction de transition de l'automate ; elle peut être partielle ;
- $F \subseteq Q$ est l'ensemble des états états accepteurs (terminaux).

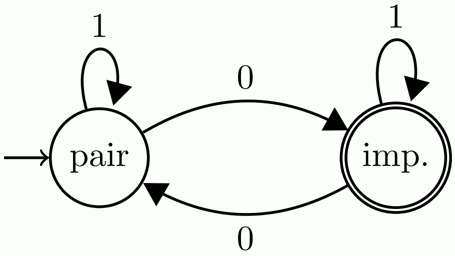
Un AD est dit complet, si sa fonction de transition est *totale*.

Automate déterministe : exemples

Considérons les mots dans $\{0,1\}^*$.

Exemple (Nombre *impair* de 0's – représentation graphique)

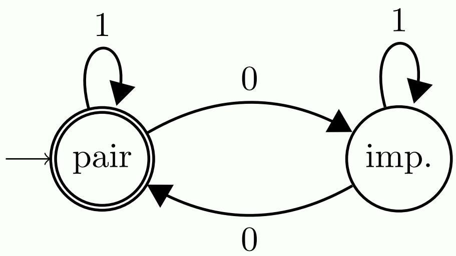
Un AD (complet) qui reconnaît les mots avec un nombre impair de 0.



- $Q = \{\text{pair}, \text{imp.}\}$
- $\Sigma = \{0, 1\}$
- $q_{\text{init}} = \text{pair}$
- $\delta = \{(\text{pair}, 0, \text{imp.}), (\text{pair}, 1, \text{pair}), (\textit{imp.}, 0, \textit{pair}), (\textit{imp.}, 1, \textit{imp.})\}$
- $F = \{\textit{imp.}\}$

Exemple (Nombre *pair* de 0's – représentation graphique)

Un AD (complet) qui reconnaît les mots avec un nombre pair de 0.



Automate déterministe : représentation tabulaire

Exemple (Nombre *impair* de 0's)

	$\downarrow$ pair	imp.*
0	imp.	pair
1	pair	imp.

Exemple (Nombre *pair* de 0's)

	$\downarrow$ pair*	imp.
0	imp.	pair
1	pair	imp.

D'autres représentations (équivalentes) existent :

- inversion lignes et colonnes,
- différents marquages des états finaux et de l'état initial.

Plan Chap. 3 - Automates déterministes

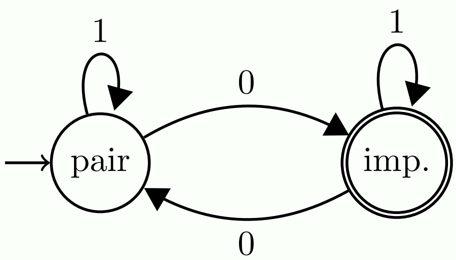
Configuration d'un automate déterministe

Dans la suite, nous considérons un AD $A = (Q, \Sigma, q_{\text{init}}, \delta, F)$.

Définition (Configuration)

Une configuration de l'automate A est un couple (q, u) où $q \in Q$ et $u \in \Sigma^*$.

Exemple (Configuration)



- $(\text{pair}, 10)$
- (pair, ϵ)
- $(\text{imp.}, \epsilon)$
- $(\text{imp.}, 000)$

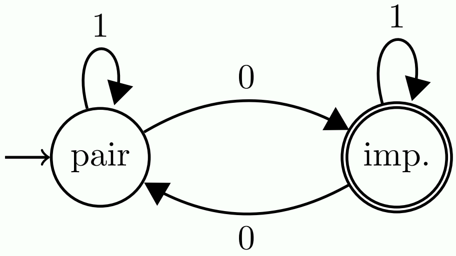
Relation de dérivation d'un automate déterministe

Définition (Relation de dérivation)

La relation de dérivation entre configurations, notée $\rightarrow$, est définie comme suit :

$$\forall q \in Q, \forall a \in \Sigma, \forall u \in \Sigma^* : (q, a \cdot u) \rightarrow (q', u) \text{ ssi } \delta(q, a) = q'.$$

Exemple (Relation de dérivation)



- $(\text{pair}, 10) \rightarrow (\text{pair}, 0)$
- $(\text{pair}, 0000) \rightarrow (\text{imp.}, 000)$
- $(\text{imp.}, \epsilon) \rightarrow$ ✗
- $(\text{imp.}, 000) \rightarrow (\text{pair}, 00)$

Exécution d'un automate déterministe

Définition (Exécution)

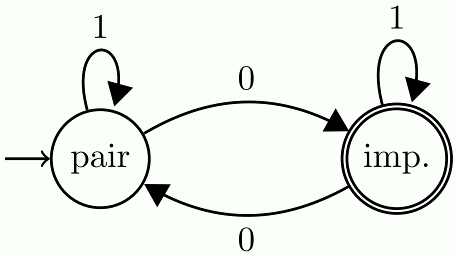
Une exécution de l'automate A est une séquence de configurations $(q_0, u_0) \cdots (q_n, u_n)$ telle que :

- $q_0 = q_{\text{init}}$,
- $\forall i \in \{0, \ldots, n-1\} : (q_i, u_i) \rightarrow (q_{i+1}, u_{i+1})$.

Exécution d'un mot

L'exécution de l'automate A sur un mot u est l'exécution avec le mot u placé dans la configuration initiale.

Exemple (Exécution d'un mot)



- Exécution de cet automate sur 10101011.

◁ Exécution de l'automate

Langage reconnu par un automate déterministe

Définition (Acceptation d'un mot par un automate)

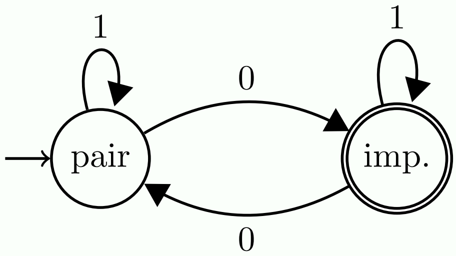
Un mot $u \in \Sigma^*$ est accepté par A, s'il existe une exécution de u sur A

$$(q_0, u_0) \cdots (q_n, u_n)$$

de A telle que :

- $u_0 = u$,
- $u_n = \epsilon$,
- $q_n \in F$.

Exemple (Acceptation d'un mot par un automate)



Mots acceptés :

- 01
- 1000
- 10111
- ...

Mots non acceptés :

- ϵ
- 11
- 1010
- ...

On vérifie que de tels mots sont acceptés ou non en déterminant leur exécution et en utilisant le critère de la définition d'acceptation.

Langage reconnu par un automate déterministe

Définition (Langage reconnu par un automate)

Le langage reconnu par A, qu'on note par $L(A)$, est l'ensemble

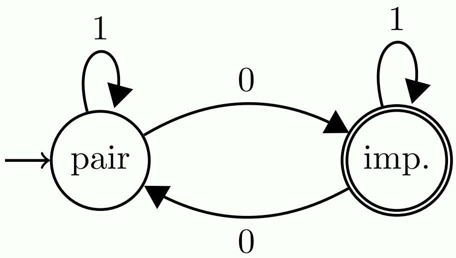
$$\{u \in \Sigma^* \mid u \text{ est accepté par } A\}.$$

Définition (Langage à états)

Un langage $L \subseteq \Sigma^*$ est appelé langage à états, s'il existe un automate déterministe qui reconnaît L.

La classe (cad l'ensemble) des langages à états est dénotée par EF.

Exemple (Langage reconnu)



Cet automate reconnaît l'ensemble des mots sur l'alphabet $\{0, 1\}$ qui contiennent un nombre impair d'occurrences du symbole 0 (quelque soit le nombre d'occurrences du symbole 1).

Cet ensemble de mots est un langage à états.

Langage reconnu par un automate déterministe : exemples/exercices

Soit $\Sigma = \{0, 1\}$.

Exercice : donner un automate qui reconnaît un langage

- Donner un automate qui accepte tous les mots qui contiennent un nombre de 0 multiple de 3.
- Donner une exécution de cet automate sur 1101010.

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

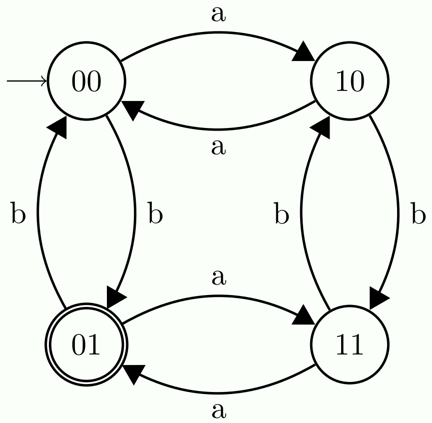
Questions

- L'ensemble des mots dans lesquels b ne précède jamais a est-il un langage à états ?
- L'ensemble des mots dans lesquels a est toujours immédiatement suivi de b est-il un langage à états ?
- L'ensemble des mots qui contiennent autant de a que de b est-il un langage à états ?

Langage reconnu par un automate déterministe : plus d'ex./exercices

Exercice : langage reconnu par un automate

Quel est le langage reconnu par l'automate suivant :



Exercice : langage à états ou non ?

Pour $k \in \mathbb{N}$, soit L_k l'ensemble des mots u tel que $|u| < k$ et u contient le même nombre de a et de b.

- L_k est-il un langage à états ?
- $\bigcup_{k \in \mathbb{N}} L_k$ est-il un langage à états ?

Plan Chap. 3 - Automates déterministes

Fonction de transition étendue aux mots

Définition et revisite des notions de mot accepté et langage reconnu

Soit $A = (Q, \Sigma, q_{\text{init}}, \delta, F)$ un AD.

Définition (Fonction de transition étendue aux mots)

À partir de δ, on définit la **fonction de transition étendue aux mots** δ^*.
Pour $q \in Q$ et $u = a_1 \cdot a_2 \cdots a_n$:

$$\delta^*(q, u) = \delta\Big(\ldots \delta\big(\delta(q, a_1), a_2\big) \ldots, a_n\Big).$$

En utilisant la définition inductive des mots.

Définition (Fonction de transition étendue aux mots - définition inductive)

À partir de δ, on définit la **fonction de transition étendue aux mots** δ^* :

- $\delta^*(q, \epsilon) = q$, pour tout état $q \in Q$,
- $\delta^*(q, w \cdot a) = \delta(\delta^*(q, w), a)$, pour tout état $q \in Q$, mot $w \in \Sigma^*$, $a \in \Sigma$.

Propriétés

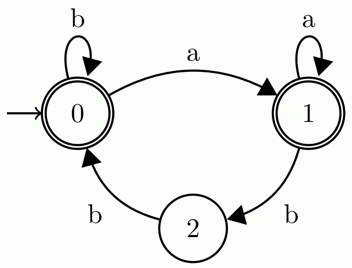
- Un mot u est accepté par A ssi $\delta^*(q_{\text{init}}, u) \in F$.
- Le langage reconnu par A est $\{u \in \Sigma^* \mid \delta^*(q_{\text{init}}, u) \in F\}$.

Fonction de transition étendue aux mots

Exemple

Soit $A = (Q, \Sigma, q_{\text{init}}, \delta, F)$ un AD.

Exemple (Fonction de transition étendue aux mots)



- $\delta^*(0, a) = 1$
- $\delta^*(0, a \cdot b) = 2$
- $\delta^*(0, a \cdot a \cdot b) = 2$
- $\delta^*(0, b \cdot a \cdot a \cdot b \cdot b) = 0$

Remarque $\delta^*(0, b \cdot a \cdot a \cdot b \cdot a)$ est non défini car $\delta(2, a)$ est non défini. □

Plan Chap. 3 - Automates déterministes

Accessibilité dans les AD : définition

Considérons un automate déterministe $A = (Q, \Sigma, q_{\text{init}}, \delta, F)$.

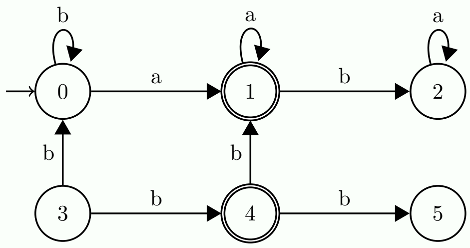
Définition (Accessibilité d'un état dans un AD)

$q \in Q$ est **accessible** dans A s'il existe un mot $u \in \Sigma^*$ tel que $\delta^*(q_{\text{init}}, u) = q$.

Définition (Co-accessibilité d'un état dans un AD)

$q \in Q$ est **co-accessible** dans A s'il existe un mot $u \in \Sigma^*$ tel que $\delta^*(q, u) \in F$.

Exemple (États accessibles et co-accessibles)

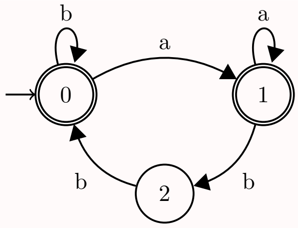


- accessibles : 0, 1, 2
- non accessibles : 3, 4, 5
- co-accessibles : 0, 1, 3, 4
- non co-accessibles : 2, 5

Remarque Nous reviendrons plus loin dans le cours sur l'accessibilité et la co-accessibilité (décidabilité et algorithmes).

Plan Chap. 3 - Automates déterministes

Résumé du chapitre : Automates Déterministes



- définition : **ensemble d'états, état initial, alphabet, fonction de transition, états accepteurs** ;
- **configuration** : couple formé par un état et un mot (à lire) ;
- **relation de dérivation** : relation entre configurations (suivant la fonction de transition) ;
- **exécution (acceptée)** : séquence de configurations (telle que la dernière configuration est formée par un état accepteur et le mot vide) obtenue en consommant le mot ;
- **langage reconnu** : ensemble des mots dont l'exécution est acceptée ;
- **langage à états** : langage qui peut être défini comme le langage reconnu d'un automate ;
- **état accessible** : état que l'on peut « atteindre » en suivant la fonction de transition.
- **état co-accessible** : état qui permet d'« atteindre » un état accepteur en suivant la fonction de transition.