Automates finis

- Automate fini déterministe (AFD)
- Langage d'un AFD
- Automates finis non déterministes AFN
- Construction d'automates
- Théorème de Kleene
- Équivalence d'automates – Minimisation

- Nous abordons une des notions les plus importantes de ce cours, celle d'automate et de langage automatique.

- Les automates fournissent un moyen pratique simple et concis de spécifier des langages complexes, leur implémentation et leur manipulation sur ordinateur est très facile.

Automates finis déterministes AFD

Idée de départ : On cherche à définir un procédé pratique permettant, efficacement, de savoir si un mot appartient à un langage L.

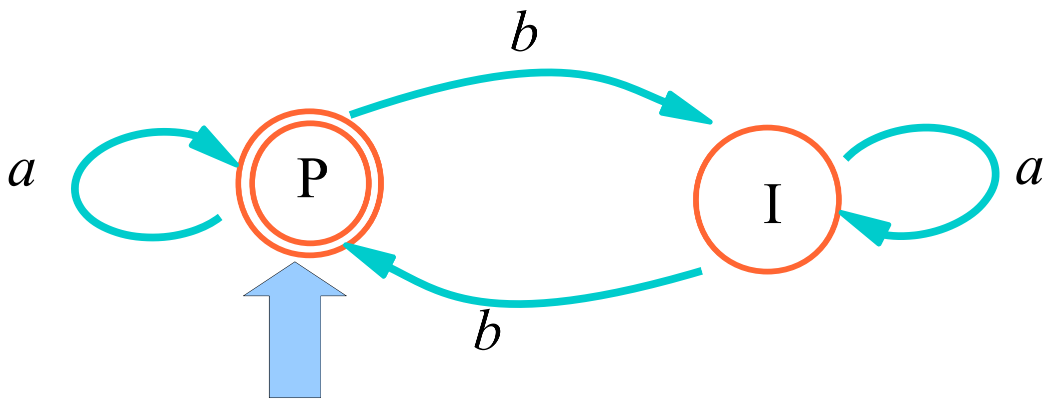
Exemple : Les mots contenant un nombre pair de « b »

Efficace : Le langage est décrit avec un minimum d'objets
Efficace : Un seul parcours du mot

Pratique : En fin de parcours, on sait si le mot est dans le langage.

Avant de donner une définition précises, remarquons que ce problème peut être modélisé à l'aide d'un « graphe » un peu particulier...

- On cherche à décrire les mots contenant un nombre pair de « b »

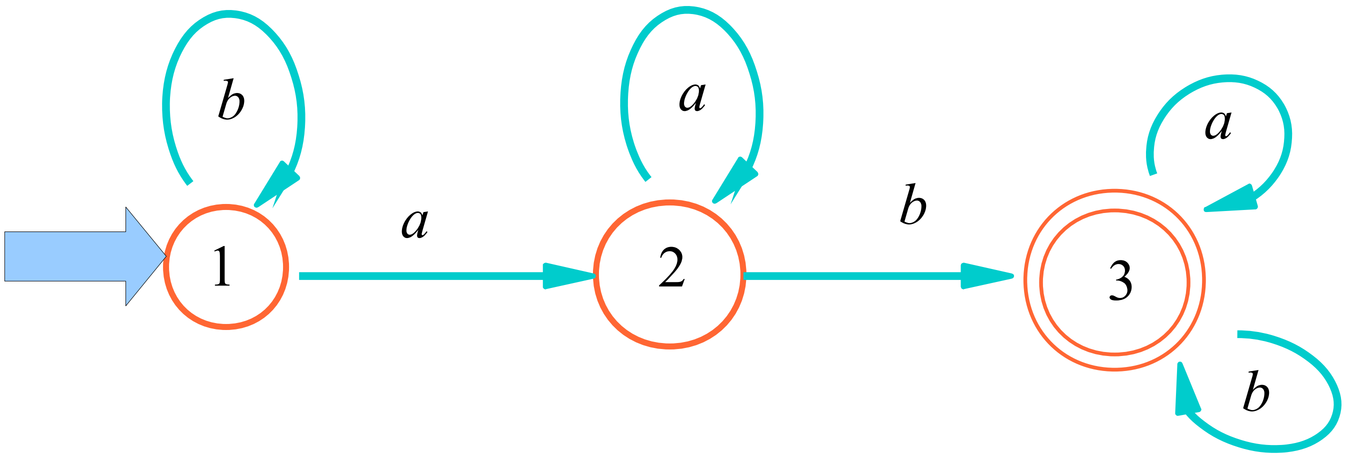


Dans l'état « P », on a lu un nombre impair de « b », dans l'état « I », on en a lu un nombre impair.

On commence dans l'état « P »

Le mot est accepté si on termine dans l'état « P »

- Décrivons le langage constitué des mots qui contiennent la chaîne « ab »



Définition

Un automate fini déterministe (AFD) est un quintuplet :

$$\mathcal{A} = (Q, \Sigma, T, q_0, A)$$

où

- Q est un ensemble fini appelé ensemble des états
- Σ est un ensemble fini appelé alphabet
- $q_0 \in Q$ est un état appelé état initial
- $A \subseteq Q$ est l'ensemble des états acceptants
- $T : Q * \Sigma \rightarrow Q$ est une fonction de transition qui à un état q et à une lettre l associe un nouvel état q'.

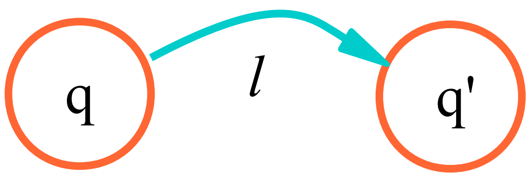
Représentation d'un AFD

(Voir l'animation AFD-Représentations)

Un automate fini déterministe peut être représenté

- Mathématiquement, par le quintuplet $\mathcal{A} = (Q, \Sigma ,T,q_0,A)$

- Par sa représentation sagittale
  - Σ doit être précisé
  - Les états sont des cercles numérotés
  - L'état initial est représenté par une grosse flèche
  - Les états acceptants sont doublement cerclés
  - La transition q'=T(q,l) est représentée par un arc



- Par une table de transition :

$Q\backslash\Sigma$	a	b
1	2	1
2	3	3
3	3	1

- Pour une implémentation sur machine, on aura donc le choix entre
  - Donner Σ, q0, A et représenter le graphe (par sa liste d'arêtes ou ses listes d'adjacence)
  - Donner Σ, q_0, A et représenter la table de transition par une matrice

Représenter graphiquement l'automate $\mathcal{A} = (Q, \Sigma, T, q_0, A)$ donné par :

- $\Sigma=\{a,b\}$
- $Q=\{1,2,3\}$
- $q_0 = 1$
- $A=\{3\}$
- T=

Q\Σ	***a***	***b***
→ 1	2	1
2	3	3
3	3	1

et en donner deux implémentations possibles en machine.

Lecture d'un mot

Soit $\mathcal{A} = (Q, \Sigma, T, q_0, A)$ un automate, on définit :

- L'état q' dérive de l'état q en 1 étape par la lecture de la lettre ℓ ssi

  $$q' = T(q, \ell)$$

  On note cette lecture :

  $$q \xrightarrow{\ell} q'$$

- L'état q' dérive de l'état q en n étape par lecture du mot $w = \ell_1 .. \ell_n$ s'il existe une suite d'états q_1, q_2, q_n telle que

  $$q \xrightarrow{\ell_1} q_1 \xrightarrow{\ell_2} q_2 \xrightarrow{\ell_3} \quad \xrightarrow{\ell_n} q_n = q'$$

Définition

On dira également que dans ce cas que :
La lecture du mot w par l'automate $\mathcal{A} = (Q, \Sigma, T, q_0, A)$ conduit de l'état q à l'état q'.

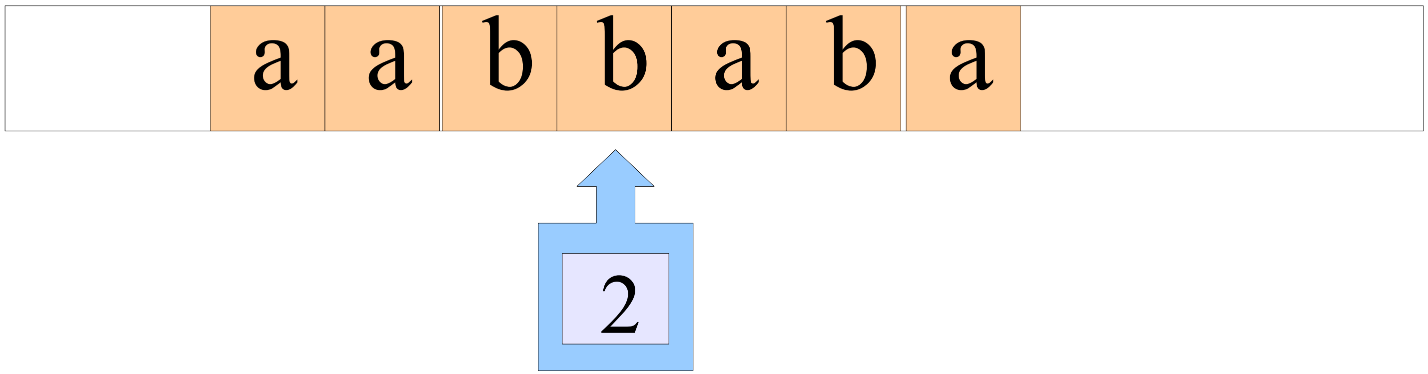
(Voir l'animation AFD-Lecture)

On peut alors donner une série de définitions fondamentales :

Définition

On dit que le mot w est accepté par l'automate $\mathcal{A} = (Q, \Sigma, T, q_0, A)$ si la lecture du mot w à partir de l'état initial q_0 conduit à un état acceptant $q \in A$.

Un automate lisant un mot w peut également être vu comme une « machine à ruban » effectuant une suite de translations sur un ruban découpé en cases dans lesquelles sont écrites des lettres de l'alphabet.



On part initialement de la première lettre du mot dans l'état q0. A chaque étape on lit la lettre écrite dans la case et la tête de lecture change d'état.

En fin de lecture, le mot est accepté si la tête de lecture est dans un état acceptant.

Une fois définis la lecture d'un mot et le langage reconnu par un automate, les questions qui se posent naturellement sont les suivantes :

- Étant donné un automate, comment « calculer » le langage qu'il accepte?
- Étant donné un langage, comment savoir s'il est automatique ou pas?
  - S'il est automatique, comment construire un automate le reconnaissant?
  - Peut-on alors construire un automate minimal reconnaissant ce langage, par exemple utilisant le moins possibles d'états?

Les paragraphes suivants vont répondre à toutes ces questions...

Langage d'un automate

Définition

On appelle langage de l'automate $\mathscr{A} = (Q, \Sigma, T, q_0, A)$ l'ensemble des mots acceptés par cet automate.

On le note $L(\mathscr{A})$.

On dit encore que $\mathscr{A}$ reconnaît ou accepte ce langage.

Définition

Un langage est dit automatique s'il existe un automate fini déterministe reconnaissant ce langage.

Définition

Deux automates sont dits équivalents si ils reconnaissent le même langage.