LifLF – Théorie des langages formels

Sylvain Brandel
2016 – 2017
sylvain.brandel@univ-lyon1.fr

CM 3

AUTOMATES À ÉTATS FINIS

Automates à états finis

- Automate ou Machine
  - *Ang. Automaton / Machine*
- Automates ou Machines
  - *Ang. Automata / Machines …*

- Automates (à états) finis déterministes
  - *Ang. Deterministic Finite Automata (DFA)*
- Automates (à états) finis non déterministes
  - *Ang. Nondeterministic Finite Automata (NFA)*
- (chapitre suivant) Automates à pile
  - *Ang. Pushdown Automata (PDA)*
- (l'an prochain) Machines de Turing
  - *Ang. Turing Machines*

Automates à états finis

- Exemple concret
  - Machine à café dans le hall du déambu
  - Barrière de péage du périph’ (pas liber-t …)
  - …

Automates finis déterministes

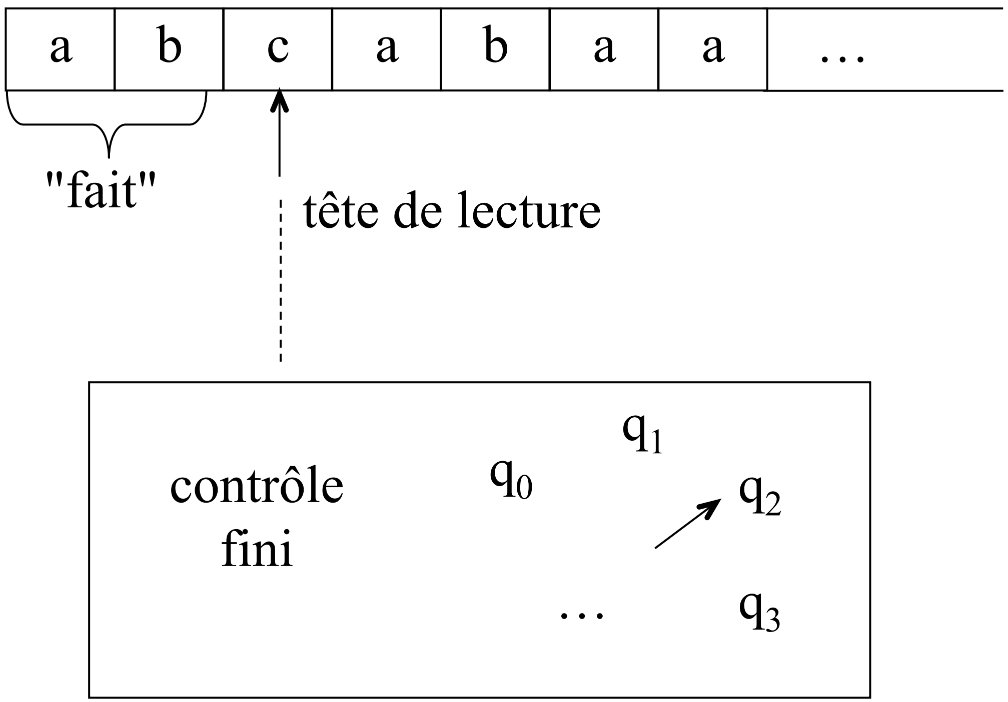
- Simulation d'une machine très simple :
  - mémorisation d'un état
  - programme sous forme de graphe étiqueté indiquant les transitions possibles
- Cette machine lit un mot en entrée.
- Ce mot décrit une suite d'actions et progresse d'état en état

  → jusqu'à la lecture complète du mot.
- Lorsque le dernier état est distingué (état final)

  → on dit que le mot est accepté.

$\Rightarrow$ Un automate permet de reconnaître un langage.

Automates finis déterministes



- Un état dépend uniquement
  - De l'état précédent
  - Du symbole lu

Automates finis déterministes

- Un automate déterministe fini est le quintuplet

  $$M = (K, \Sigma, \delta, s, F) \text{ où :}$$

  - K : ensemble fini (non vide) d'états
  - Σ : alphabet (ensemble non vide de lettres)
  - δ : fonction de transition : $K \times \Sigma \rightarrow K$

    $\delta(q, \sigma) = q'$ (q' : état de l'automate après avoir lu la lettre σ dans l'état q)
  - s : état initial : $s \in K$
  - F : ensemble des états finaux : $F \subset K$

Automates finis déterministes

- <u>Exécution</u>

a	b	c	a	b	a	a	…

La machine

- lit a (qui est ensuite oublié),
- passe dans l'état $\delta(s, a)$ et avance la tête de lecture,
- répète cette étape jusqu'à ce que tout le mot soit lu.

- La partie déjà lue du mot ne peut pas influencer le comportement à venir de l'automate.

→ d'où la notion de <u>configuration</u>

Automates finis déterministes

- <u>Configuration</u>
  - état dans lequel est l'automate
  - mot qui lui reste à lire (partie droite du mot initial)

- Formellement : une configuration est un élément quelconque de $K \times \Sigma^*$.

- <u>Exemple</u>

  sur l'exemple précédent, la configuration est (q_2, cabaa).

Automates finis déterministes

- Le fonctionnement d'un automate est décrit par le passage d'une configuration à une autre, cette dernière obtenue
  - en lisant un caractère,
  - et en appliquant la fonction de transition.

- <u>Exemple</u>
  - $(q_2, \text{cabaa}) \rightarrow (q_3, \text{abaa})$ si $\delta(q_2, c) = q_3$

Automates finis déterministes

- Un automate M détermine une relation <u>binaire</u> entre configurations qu'on note $\vdash_M$ définie par :
  - $\vdash_M \subset (K \times \Sigma^*)^2$
  - $(q, w) \vdash_M (q', w')$ ssi $\exists\, a \in \Sigma$ tel que $w = aw'$ et $\delta(q, a) = q'$

- On dit alors que <u>on passe de</u> (q, w) <u>à</u> (q', w') <u>en une étape</u>.

Automates finis déterministes

- On note $\vdash_M^*$ la fermeture transitive réflexive de $\vdash_M$:
  (q, w) $\vdash_M^*$ (q', w') signifie qu'on passe de (q, w) à (q', w') en zéro, une ou plusieurs étapes

- Un mot w est <u>accepté</u> par M
  ssi (s, w) $\vdash_M^*$ (q, e), avec $q \in F$.

- Le <u>langage accepté</u> par M est l'ensemble de tous les mots acceptés par M.
  Ce langage est noté L(M).

Automates finis déterministes

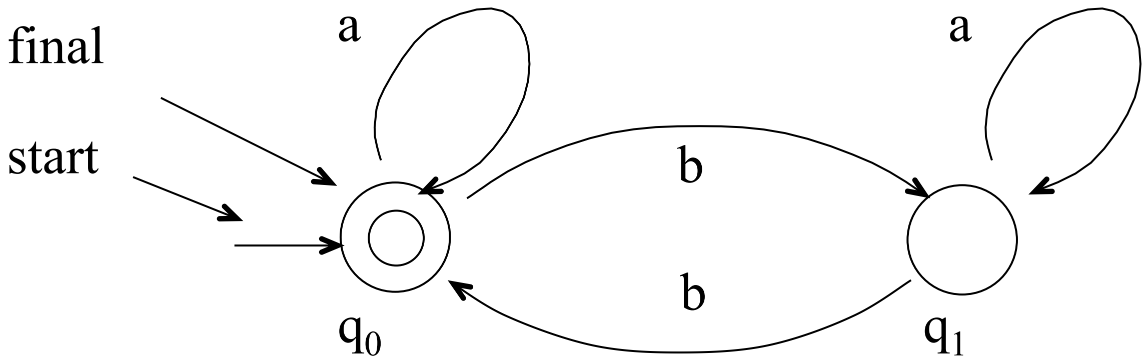
- <u>Exemple</u> M = (K, Σ, δ, s, F)
  - K = {q_0, q_1}
  - Σ = {a, b}
  - s = q_0
  - F = {q_0}

δ :

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

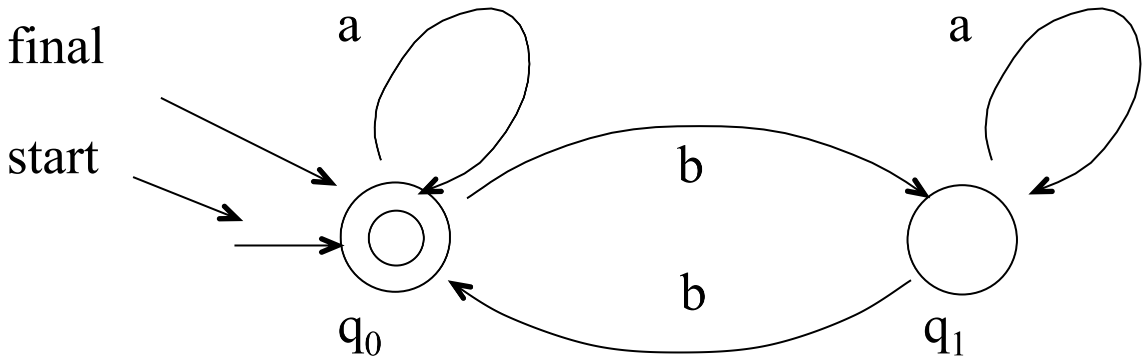
$\Leftrightarrow$

	a	b
q_0	q_0	q_1
q_1	q_1	q_0



Automates finis déterministes

- Démo

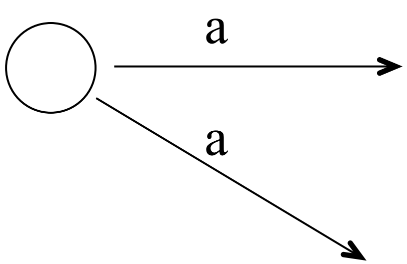


Automates finis non déterministes

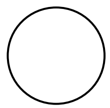
- Idée : remplacer la fonction $\vdash_M$ (ou δ) par une relation.
- Une relation, c'est beaucoup plus général qu'une fonction.

→ on a ainsi une classe plus large d'automates.

⇒ Dans un état donné, on pourra avoir :



ou

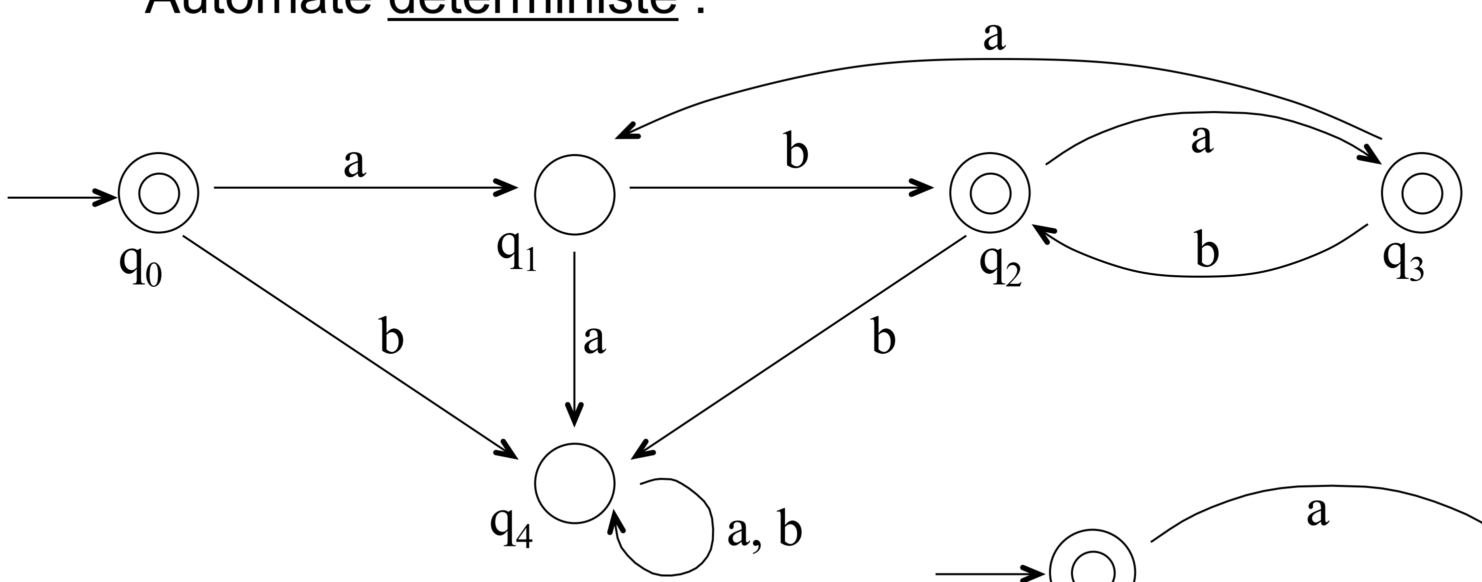


(pas de a-transition)

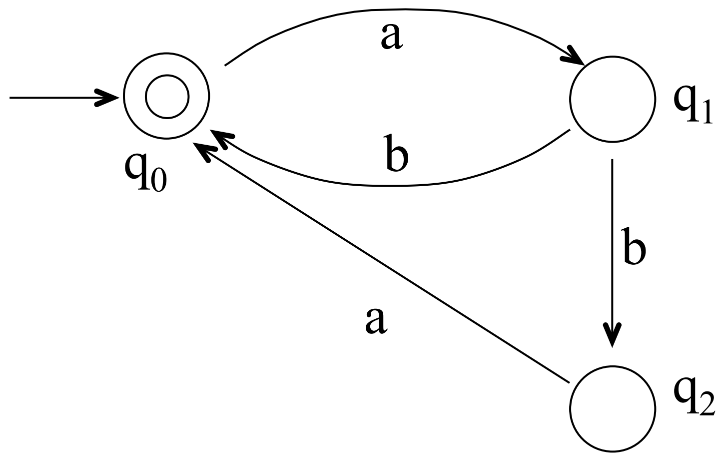
Automates finis non déterministes

- $L = (ab \cup aba)^*$

Automate <u>déterministe</u> :



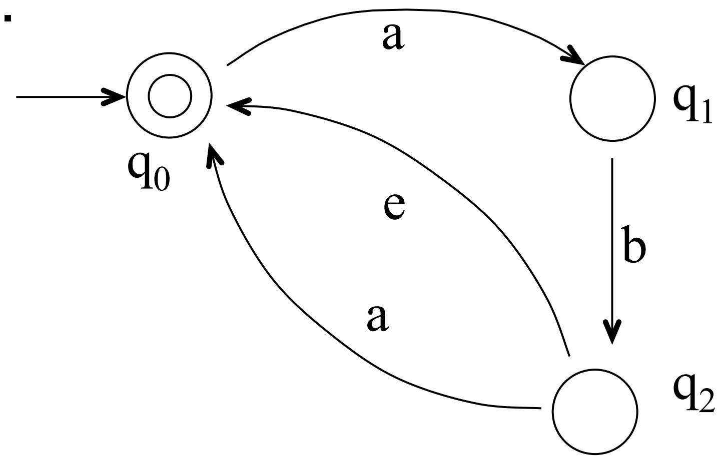
Automate <u>non déterministe</u> :



Automates finis non déterministes

- Dans le cas de l'automate non déterministe, un mot est <u>accepté</u> s'il existe <u>un ou plusieurs chemins</u> (au moins un) pour aller de l'état initial (ici q_0) à l'état final (ici q_0).
- <u>Autre différence</u> par rapport aux automates déterministes :
  – Il est possible d'étiqueter une flèche par le symbole e.

- Autre formulation (encore plus intuitive) de l'automate précédent :



Automates finis non déterministes

- Un automate non déterministe fini est le quintuplet

  $M = (K, \Sigma, \Delta, s, F)$ où :

  - K : ensemble fini (non vide) d'états
  - Σ : alphabet (ensemble non vide de lettres)
  - Δ : relation de transition : $K \times (\Sigma \cup \{e\}) \times K$

    $(q, \sigma, p) \in \Delta$: σ-transition ($\sigma \in \Sigma$)
  - s : état initial : $s \in K$
  - F : ensemble des états finaux : $F \subset K$

(hormis la relation, le reste est identique à la formulation déterministe)

Automates finis non déterministes

- Si $(q, e, p) \in \Delta$: on a une ε-transition (transition spontanée)

  → On passe de q à p sans lire de symbole dans le mot courant.

- Une configuration est un élément de $K \times \Sigma^*$.

- M décrit une relation binaire entre configurations qu'on note $\vdash_M$:

  $(q, w) \vdash_M (q', w')$

  ssi il existe un mot d'au plus une lettre $u \in \Sigma \cup \{e\}$

  tq $w = uw'$ et $(q, u, q') \in \Delta$

Automates finis non déterministes

- $\vdash_M$ est une <u>relation</u> et non plus une fonction (automates déterministes) :
  - (q, e) peut être en relation avec une autre configuration (après une ε-transition)
  - pour une configuration (q, w), il peut y avoir plusieurs configurations (q', w') (ou aucune) tq (q, w) $\vdash_M$ (q', w')

- On note comme avant $\vdash_M^*$ la fermeture transitive réflexive de $\vdash_M$

- Un mot w est <u>accepté</u> par M ssi (s, w) $\vdash_M^*$ (q, e) avec q $\in$ F.

- L(M) est le langage de tous les mots acceptés par M.

Et maintenant ?

- Déterministe ou non déterministe ?
  - Même classe de langages reconnus
  - Donc équivalents
- Langages rationnels et langages reconnus par les automates finis ?
  - Même classe de langages
- Un langage est-il rationnel ?
  - Preuve de rationalité
- Un automate est-il minimal ?
  - Automate standard