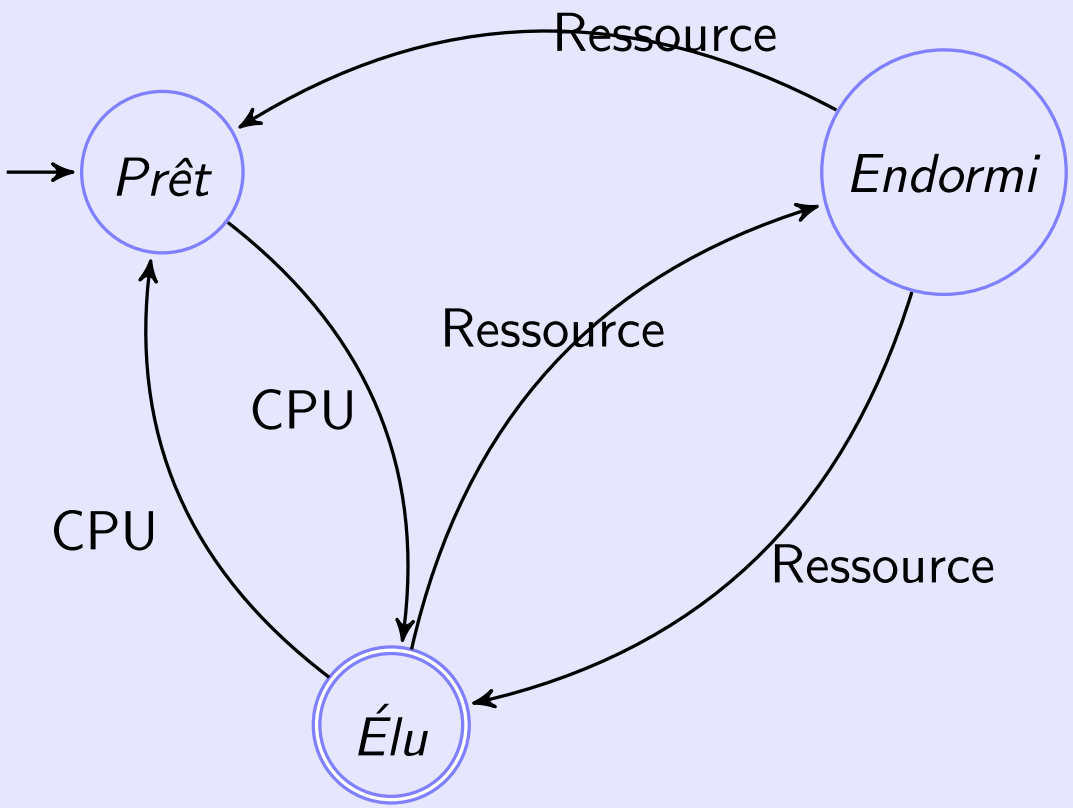
Chapter 12
Automata

Discrete Structures for Computing on 27 May 2014

Huynh Tuong Nguyen, Tran Vinh Tan
Faculty of Computer Science and Engineering
University of Technology - VNUHCM

Contents

États standards d'un prosessus



- O avec label: états
- $\rightarrow$: transitions

Alphabets, symboles

Définition

Un *alphabet* Σ est un ensemble fini dont les éléments sont appelés symboles (ou caractères).

Remarque

Σ sera prèsque toujours l'ensemble des caractères accessibles (lettres minuscules, lettres majuscules, chiffres, symboles et caractères spéciaux comme l'espace ou le retour à la ligne), mais rien n'empêche d'imaginer d'autres ensembles.

Mots

Définition

- Un mot u sur Σ est une chaine finie (éventuellement vide) de symboles (ou de caractères) dans Σ.
- Le mot vide est noté par ε.
- La longueur du mot, notée par $|u|$, est le nombre de ses caractères.
- L'ensemble des mots sur Σ sera noté Σ^*.
- Un langage L sur Σ est un sous-ensemble de Σ^*.

Remarque

Le but de notre machine est d'analyser un mot de Σ^* pour savoir si celui-ci appartient ou non à L.

Exemple

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

- ε est le mot de longueur 0.
- 0 et 1 sont les mots de longueur 1.
- 00, 01, 10 et 11 sont les mots de longueur 2.
- $\emptyset$ est un langage sur Σ. Il est appelé le langage vide.
- Σ^* est un langage sur Σ. Il est appelé le langage universel.
- $\{\varepsilon\}$ est un langage sur Σ.
- $\{0, 00, 001\}$ est aussi un langage sur Σ.
- L'ensemble des mots qui contiennent un nombre impair de 0 est un langage sur Σ.
- L'ensemble des mots qui contiennent autant de 0 que de 1 est un langage sur Σ.

Concaténation de mots

Intuitivement, la concaténation des mots 01 et 10 est le mot 0110.
La concaténation du mot vide ε et le mot 110 est le mot 110.

Définition

La concaténationest une application de $\Sigma^* \times \Sigma^*$ vers Σ^*.
La concaténation de deux mots u et v dans Σ est le mot $u.v$.

Huynh Tuong Nguyen,
Tran Vinh Tan

Expressions régulières

Permet de spécifier un langagepar des chaines de caractères constituées de lettres et de ε, de parenthèses $()$, de symboles opératoires $+$, $.$, $*$. Cette chaîne peut être vide, notée $\emptyset$.

Opérations régulières sur les langages

- réunion ensembliste $\cup$ ou $+$
- produit de concaténation
- fermeture transitive $*$

Opérations régulières sur les langages

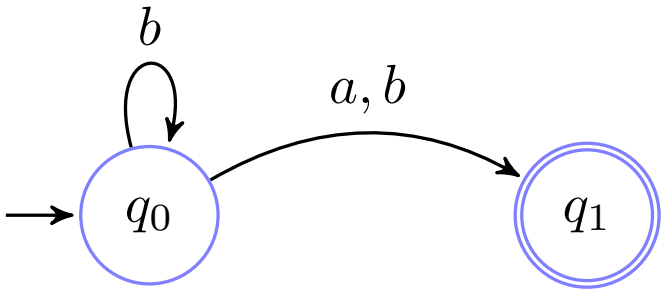
- $(a+b)^*$ représente tous les mots sur l'aphabet $\Sigma = \{a, b\}$
- $a^*(ba^*)^*$ représente le même langage
- $(a+b)^*aab$ représente les mots se terminant par aab.

Automate fini

- Le but est de représentation d'un système de processus.
- Il consiste des états (incluant un état initial et un ou plussieurs états finaux/acceptant) et des transitions (événement).
- Le nombre des états doit être fini.

Automate fini

- Le but est de représentation d'un système de processus.
- Il consiste des états (incluant un état initial et un ou plussieurs états finaux/acceptant) et des transitions (événement).
- Le nombre des états doit être fini.



Expression régulière

$b^*(a+b)$

Automates finis non déterministes

Définition

Un automates finis nondéterministes (**NFA**, en anglais) est donné par un quintuplet $(Q, \Sigma, q_0, \delta, F)$ où

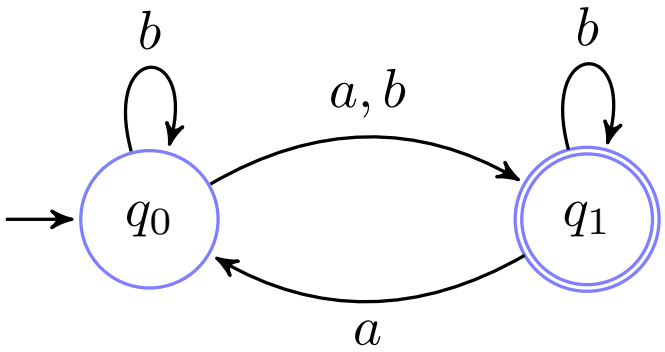
- Q est un ensemble fini d'états.
- Σ est l'alphabet de l'automate.
- $q_0 \in Q$ est l'état initial.
- $\delta : Q \times \Sigma \to Q$ est la fonction de transition.
- $F \subseteq Q$ est l'ensemble des états finaux.

Remarque

Selon un événement, un état peut-être arivé à un ou plusieurs états.

Exercice

Donner l'expression régulière pour l'automate fini suivant.

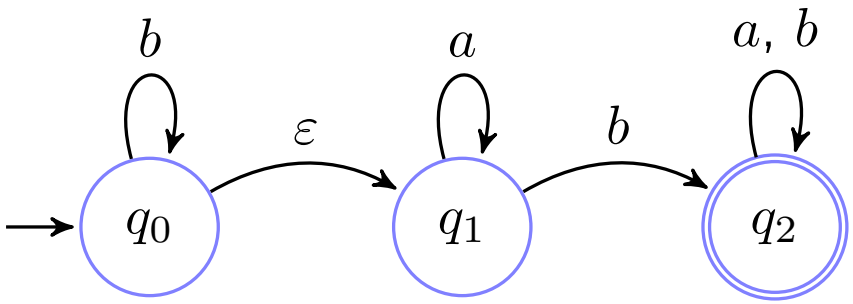


Huynh Tuong Nguyen,
Tran Vinh Tan

NFA avec symbole vide ε

Autre définition de NFA

Automate fini avec les transitions définies par caractère x (dans Σ) ou caractère vide ε.



Automates finis déterministes

Définition

Un automates finis déterministes (**DFA**, en anglais) est donné par un quintuplet $(Q, \Sigma, q_0, \delta, F)$ où

- Q est un ensemble fini d'états.
- Σ est l'alphabet de l'automate.
- $q_0 \in Q$ est l'état initial.
- $\delta : Q \times \Sigma \to Q$ est la fonction de transition.
- $F \subseteq Q$ est l'ensemble des états finaux.

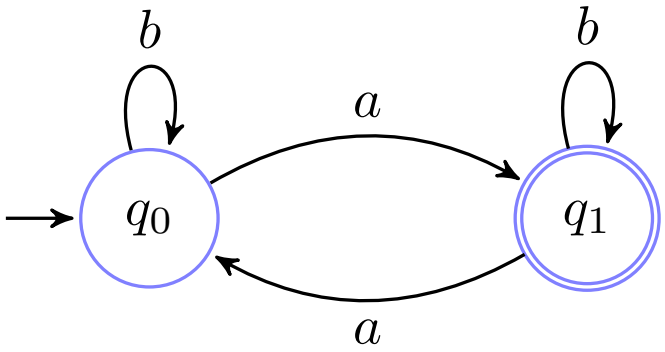
Condition

Fonction de transition δ est une application.

Exemple

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

Ci-desous un automate déterministe et complet qui reconnaît l'ensemble des mots qui contiennent un nombre impair de b.



- $Q = \{q_0, q_1\}$,
- $\delta(q_0, a) = q_1$, $\delta(q_0, b) = q_0$, $\delta(q_1, a) = q_0$, $\delta(q_1, b) = q_1$,
- $F = \{q_1\}$.

Configuration et exécutions

Soit $A = (Q, \Sigma, q_0, \delta, F)$

Une configuration de l'automate A est un couple (q, u) où $q \in Q$ et $u \in \Sigma$.
On définit la relation $\rightarrow$ de dérivation entre configurations :
$(q, a.u) \rightarrow (q', u)$ ssi $\delta(q, a) = q'$

Une exécution de l'automate A est une séquence de configurations $(q_0, u_0) \dots (q_n, u_n)$ telle que
$(q_i, u_i) \rightarrow (q_{i+1}, u_{i+1})$, pour $i = 0, 1, \dots, n-1$.

Exercice

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

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

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

Donner un automate qui accepte tous les mots qui contiennent 2 caractères a.
Donnez une exécution de cet automate sur $aabb$, $ababb$ et $bbaa$.

Langages reconnaissables

Définition

Un langage L sur un alphabet Σ, défini comme un sous-ensemble de Σ^*, est reconnaissable s'il est accepté par un automate fini.

Proposition

Si L_1 et L_2 sont deux langages reconnaissables alors

- $L_1 \cup L_2$ et $L_1 \cap L_2$ sont aussi reconnaissables;
- $L_1.L_2$ et L_1^* sont aussi reconnaissables.

Exemple

Sous-mot ab

Construire un automate fini déterministe (DFA) qui reconnait le langage des mots sur l'alphabet $\{a, b\}$ qui contiennent le sous-mot ab.

Expression régulière

$(a|b)^*ab(a|b)^*$

Exemple

Sous-mot ab

Construire un automate fini déterministe (DFA) qui reconnait le langage des mots sur l'alphabet $\{a, b\}$ qui contiennent le sous-mot ab.

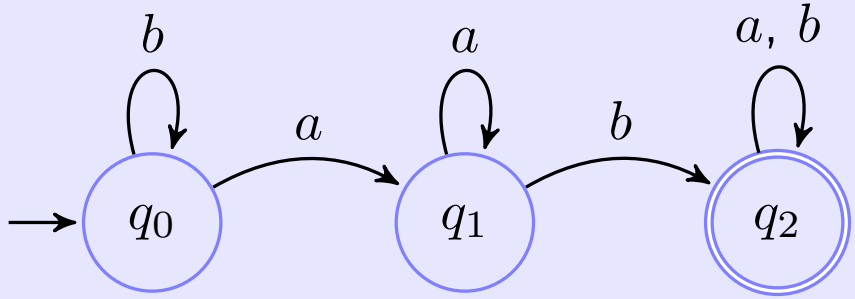
Expression régulière

$(a|b)^*ab(a|b)^*$

Autre représentation

	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
q_2	q_2	q_2

Automate



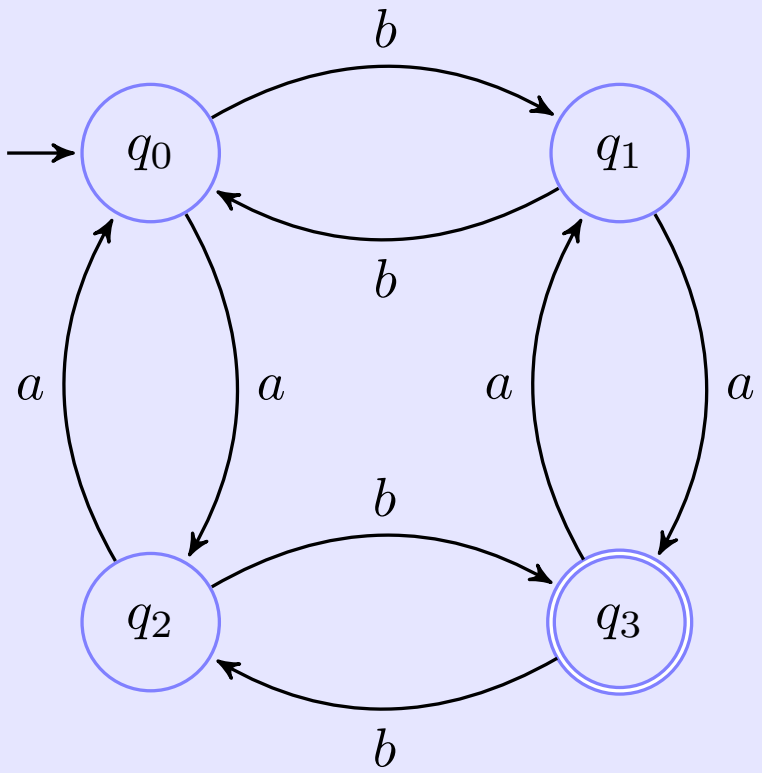
Exemple

Déterminer expression régulière et construire un DFA qui reconnait le langage des mots sur l'alphabet $\{a, b\}$ avec un nombre pair de a et un nombre pair de b.

Exemple

Déterminer expression régulière et construire un DFA qui reconnait le langage des mots sur l'alphabet $\{a, b\}$ avec un nombre pair de a et un nombre pair de b.

Automate



Autre représentation

	a	b
$\rightarrow q_0$	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

de NFA à DFA