Langage rationnel
Automate Fini Déterministe

Aspects Théoriques de l'Informatique
Licence 3 informatique

Sébastien Verel
verel@univ-littoral.fr
http://www-lisic.univ-littoral.fr/~verel

Université du Littoral Côte d'Opale
Laboratoire LISIC
Equipe OSMOSE

Objectifs de la séance 02

- Connaitre les opérations algébriques sur les langages
- Savoir définir langage rationnel
- Savoir définir le langage reconnu par un automate fini déterministe
- Savoir construire un automate fini déterministe reconnaissant un langage
- Savoir écrire l'expression régulière du langage reconnu par un automate fini déterministe

Objectifs de la séance 02

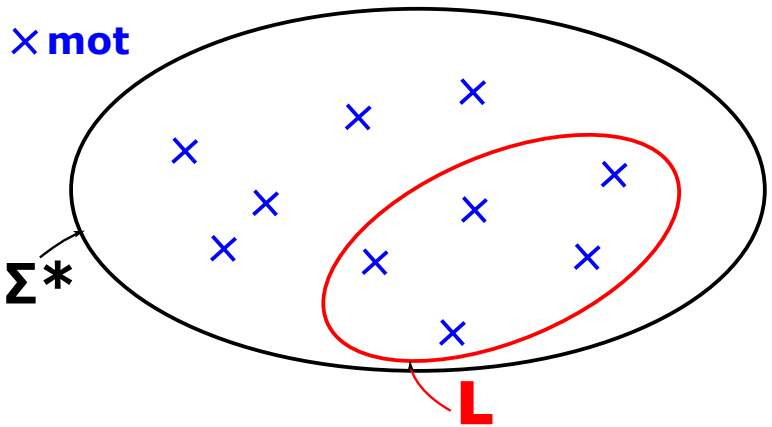
- Connaitre les opérations algébriques sur les langages
- Savoir définir langage rationnel
- Savoir définir le langage reconnu par un automate fini déterministe
- Savoir construire un automate fini déterministe reconnaissant un langage
- Savoir écrire l'expression régulière du langage reconnu par un automate fini déterministe

Question principale du jour :

Quelle langue parle mon automate ?

Plan

Définir un langage



Σ Ensemble (fini) de lettres

Σ^* Ensemble (dénombrable) des mots

Spécification de langage

- De manière informelle :
  mouais, pas très informatique, ambiguïté, à oublier.
- Par une **propriété caractéristique** P :

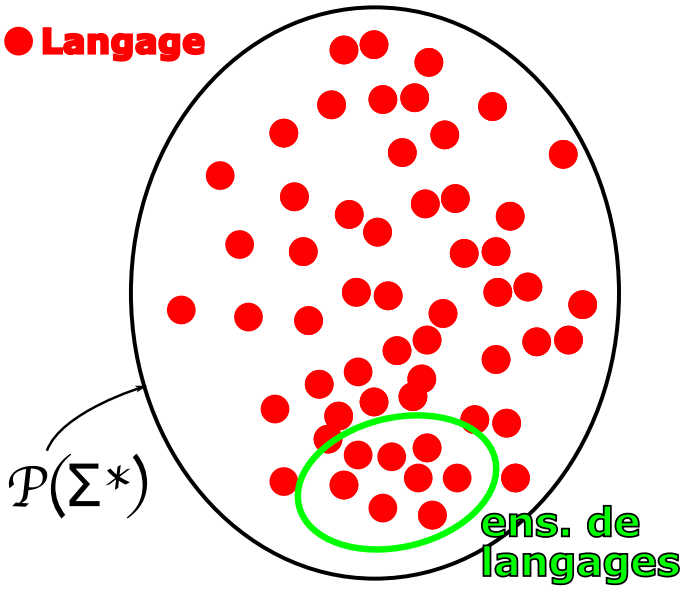
$$L_P = \{u \in \Sigma^* \ : \ P(u) \text{ vraie}\}$$

  Pas toujours un moyen pratique de déterminer si $u \in L$
- Par un algorithme A (**langage décidable**) :

$$L_A = \{u \in \Sigma^* \ : \ A(u) \text{ accept}\}$$

  Temps fini, moyen pratique pour savoir si $u \in L$

Ensemble de langages



Σ Ensemble (fini) de lettres

Σ^* Ensemble (dénombrable) des mots

$\mathcal{P}(\Sigma^*)$ Ensemble de tous les langages

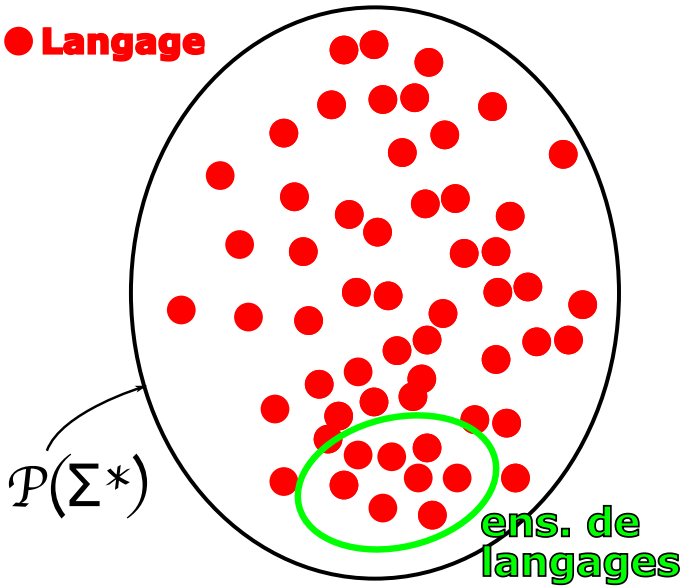
Ensemble de propriétés P définissent un ensemble de langage :

$$LP = \{L \in \mathcal{P}(\Sigma^*) \ : \ \exists P \text{ telle que } L = L_P\}$$

Ensemble d'algorithmes A définit un ensemble de langage :

$$LM = \{L \in \mathcal{P}(\Sigma^*) \ : \ \exists A \text{ telle que } L = L_A\}$$

Ensemble de langages



Σ Ensemble (fini) de lettres

Σ^* Ensemble (dénombrable) des mots

$\mathcal{P}(\Sigma^*)$ Ensemble de tous les langages

Ensemble de propriétés P définissent un ensemble de langage :

$$LP = \{L \in \mathcal{P}(\Sigma^*) \ : \ \exists P \text{ telle que } L = L_P\}$$

Ensemble d'algorithmes A définit un ensemble de langage :

$$LM = \{L \in \mathcal{P}(\Sigma^*) \ : \ \exists A \text{ telle que } L = L_A\}$$

Tout l'art consiste à définir les bonnes propriétés et les bons modèles de machines...

Cardinalité de l'ensemble des langages

Cardinalité de $\mathcal{P}(\Sigma^*)$

L'ensemble des langages $\mathcal{P}(\Sigma^*)$ sur un alphabet Σ est non dénombrable.

Cardinalité de l'ensemble des langages

Cardinalité de $\mathcal{P}(\Sigma^*)$

L'ensemble des langages $\mathcal{P}(\Sigma^*)$ sur un alphabet Σ est non dénombrable.

Preuve :
Utilisation du procédé diagonal de Cantor avec la fonction caractéristique des langages.

Cardinalité de l'ensemble des langages

Cardinalité de $\mathcal{P}(\Sigma^*)$

L'ensemble des langages $\mathcal{P}(\Sigma^*)$ sur un alphabet Σ est non dénombrable.

Preuve :
Utilisation du procédé diagonal de Cantor avec la fonction caractéristique des langages.

Les conséquences sont **graves** !...

Expressions régulières (Rappel)

Rappel

- Les expressions régulières sont les expressions que l'on peut construire à partir de $+$, . et $*$ (cf. grep)

Conséquence

Une expression régulière définit un langage.

Par exemple

- $ab(a+b+c)$ est l'ensemble $\{aba, abb, abc\}$
- $(ab)^3$ est l'ensemble $\{ababab\}$
- $(ab)^*$: tous les mots qui répétent un nombre fini de fois ab (voir nul) $\{\epsilon, ab, abab, ababab, \ldots\}$
- $ab(a+b+c)^*$: tous les mots qui comment par ab

Opérations rationnelles sur les langages

Soient deux langages L et M respect. sur les alphabets Σ_L et Σ_M. L et M sont des ensembles, il est possible d'appliquer les opérations ensemblistes.

- **union** et **intersection** : (langages sur l'alphabet $\Sigma_L \cup \Sigma_M$)

$$L \cup M \text{ et } L \cap M$$

Opérations rationnelles sur les langages

Soient deux langages L et M respect. sur les alphabets Σ_L et Σ_M. L et M sont des ensembles, il est possible d'appliquer les opérations ensemblistes.

- **union** et **intersection** : (langages sur l'alphabet $\Sigma_L \cup \Sigma_M$)

$$L \cup M \text{ et } L \cap M$$

- **complémentaire** :

$$L^c = \Sigma_L^* - L$$

Opérations rationnelles sur les langages

Soient deux langages L et M respect. sur les alphabets Σ_L et Σ_M. L et M sont des ensembles, il est possible d'appliquer les opérations ensemblistes.

- **union** et **intersection** : (langages sur l'alphabet $\Sigma_L \cup \Sigma_M$)

$$L \cup M \text{ et } L \cap M$$

- **complémentaire** :

$$L^c = \Sigma_L^* - L$$

- **produit** (langage sur l'alphabet $\Sigma_L \cup \Sigma_M$) :

$$LM = \{uv \mid u \in L \ , \ v \in M\}$$

- **puissance** :
  - $L^0 = \{\epsilon\}$
  - pour $i > 0$, $L^i = L.L^{i-1}$

Opérations rationnelles sur les langages

Soient deux langages L et M respect. sur les alphabets Σ_L et Σ_M. L et M sont des ensembles, il est possible d'appliquer les opérations ensemblistes.

- **union** et **intersection** : (langages sur l'alphabet $\Sigma_L \cup \Sigma_M$)

$$L \cup M \text{ et } L \cap M$$

- **complémentaire** :

$$L^c = \Sigma_L^* - L$$

- **produit** (langage sur l'alphabet $\Sigma_L \cup \Sigma_M$) :

$$LM = \{uv \mid u \in L \ , \ v \in M\}$$

- **puissance** :
  - $L^0 = \{\epsilon\}$
  - pour $i > 0$, $L^i = L.L^{i-1}$
- **étoile** :

$$L^* = \cup_{i \geq 0} L^i$$

Exemples

Soient $L = \{pl, dl, pdpl, ldl\}$ et $M = \{\epsilon, ai, ia\}$

- $L \cup M =$

Exemples

Soient $L = \{pl, dl, pdpl, ldl\}$ et $M = \{\epsilon, ai, ia\}$

- $L \cup M = \{pl, dl, pdpl, ldl, \epsilon, ai, ia\}$
- $L \cap M =$

Exemples

Soient $L = \{pl, dl, pdpl, ldl\}$ et $M = \{\epsilon, ai, ia\}$

- $L \cup M = \{pl, dl, pdpl, ldl, \epsilon, ai, ia\}$
- $L \cap M = \emptyset$
- $L^c = \Sigma^* - L$

Exemples

Soient $L = \{pl, dl, pdpl, ldl\}$ et $M = \{\epsilon, ai, ia\}$

- $L \cup M = \{pl, dl, pdpl, ldl, \epsilon, ai, ia\}$
- $L \cap M = \emptyset$
- $L^c = \Sigma^* - L$: trop long....
- $LM =$

Exemples

Soient $L = \{pl, dl, pdpl, ldl\}$ et $M = \{\epsilon, ai, ia\}$

- $L \cup M = \{pl, dl, pdpl, ldl, \epsilon, ai, ia\}$
- $L \cap M = \emptyset$
- $L^c = \Sigma^* - L$: trop long....
- $LM = \{pl, dl, pdpl, ldl,$
  $plai, dlai, pdplai, ldlai,$
  $plia, dlia, pdplia, ldlia, \}$
- $M^2 =$

Exemples

Soient $L = \{pl, dl, pdpl, ldl\}$ et $M = \{\epsilon, ai, ia\}$

- $L \cup M = \{pl, dl, pdpl, ldl, \epsilon, ai, ia\}$
- $L \cap M = \emptyset$
- $L^c = \Sigma^* - L$: trop long....
- $LM = \{pl, dl, pdpl, ldl,$
  $plai, dlai, pdplai, ldlai,$
  $plia, dlia, pdplia, ldlia, \}$
- $M^2 = \{\epsilon, ai, ia, aiai, iaia, aiia, iaai\}$
- étoile :

$$M^* = \cup_{i \geq 0} M^i$$

...

Langage rationnel

Définition inductive de l'ensemble des langages rationnels (ou réguliers) :

Langages rationnels (ou réguliers)

L'ensemble des langages rationnels LR est défini par :

- *base :*
  - $\emptyset \in LR$
  - $\{\epsilon\} \in LR$
  - pour tout $a \in \Sigma$, $\{a\} \in LR$.
- *induction :* Si $L \in LR$ et $M \in LR$ alors :
  - $L \cup M \in LR$
  - $L.M \in LR$
  - $L^* \in LR$.

Remarque :
LR est clos par réunion, concaténation et $*$.

Expression Régulière (ER) et Langage Rationnel (LR)

Théorème (admis)

Un langage est rationnel (ou régulier)
si et seulement si
il est définit par une expression régulière.

d'où le nom de langage rationnel / régulier...

Introduction

Spécifier un langage L :

- donner une description des mots qui sont dans L (ER, etc.),

ou, de façon équivalente :

- donner un algorithme permettant de décider si un mot quelconque est, ou n'est pas dans L.

Les automates sont des **machines abstraites** programmables qui permettent de reconnaitre si un mot appartient, ou non, à un langage L.

Ils existent dans de nombreuses machines concrètes (machine à café, robots, ...)

Exemple

Les scores au tennis sont : 15-0, 0-15, 30-0, 0-30, 40-0, ..., puis égalité, avantage A, avantage B, jeu A, jeu B.

Soit l'alphabet $\Sigma = \{a, b\}$ qui signifie que le joueur A ou B a marqué un point.

Exemple

Les scores au tennis sont : 15-0, 0-15, 30-0, 0-30, 40-0, ..., puis égalité, avantage A, avantage B, jeu A, jeu B.

Soit l'alphabet $\Sigma = \{a, b\}$ qui signifie que le joueur A ou B a marqué un point.

Quels sont les mots qui représente une succession valide des points ?

Exemple

Les scores au tennis sont : 15-0, 0-15, 30-0, 0-30, 40-0, ..., puis égalité, avantage A, avantage B, jeu A, jeu B.

Soit l'alphabet $\Sigma = \{a, b\}$ qui signifie que le joueur A ou B a marqué un point.

Quels sont les mots qui représente une succession valide des points ?

Il est possible d'associer à chaque score possible un état, puis il faut alors construire une table de transition en fonction du point marqué (c'est-à-dire de la lettre lue)

$$T : Q \times \Sigma \to Q$$

L'ensemble des états et les transitions constituent le coeur de l'automate.

Définition

Automate fini déterministe

Un **automate fini déterministe** (AFD) est un quintuplet (Q, Σ, T, q_0, A) où :

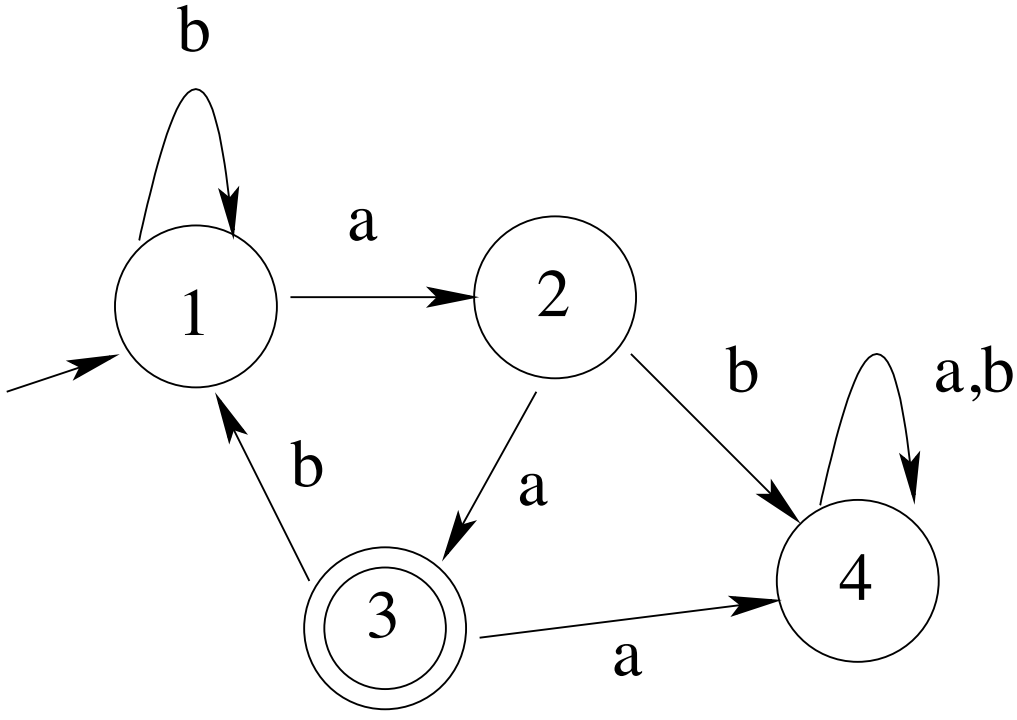
- Σ est l'*alphabet* de l'automate,
- Q un ensemble **fini** appelé *ensemble des états* de l'automate,
- T est une application de $Q \times \Sigma$ dans Q, appelée la *fonction de transition*
- q_0 est un élément de Q, appelé l'*état initial*
- A est un sous-ensemble de Q, appelé l'ensemble des *états acceptants*.

Exemples

$\Sigma = \{a, b\}$,
$Q = \{1, 2, 3, 4\}$,
$q_0 = 1$ est l'état initial
$A = \{3\}$ est l'ensemble des états acceptants.

T

		a	b
$\rightarrow$	1	2	1
	2	3	4
	$\boxed{3}$	4	1
	4	4	4



Calcul sur automate : Lecture

fonction de transition itérée

La **fonction de transition itérée** est l'application $T^* : Q \times \Sigma^* \to Q$ définie par :

- *base :* si $w = \epsilon$ alors $T^*(q, w) = q$
- *induction :* si $w = w_0 x$ avec $x \in \Sigma$ alors $T^*(q, w) = T(T^*(q, w_0), x)$

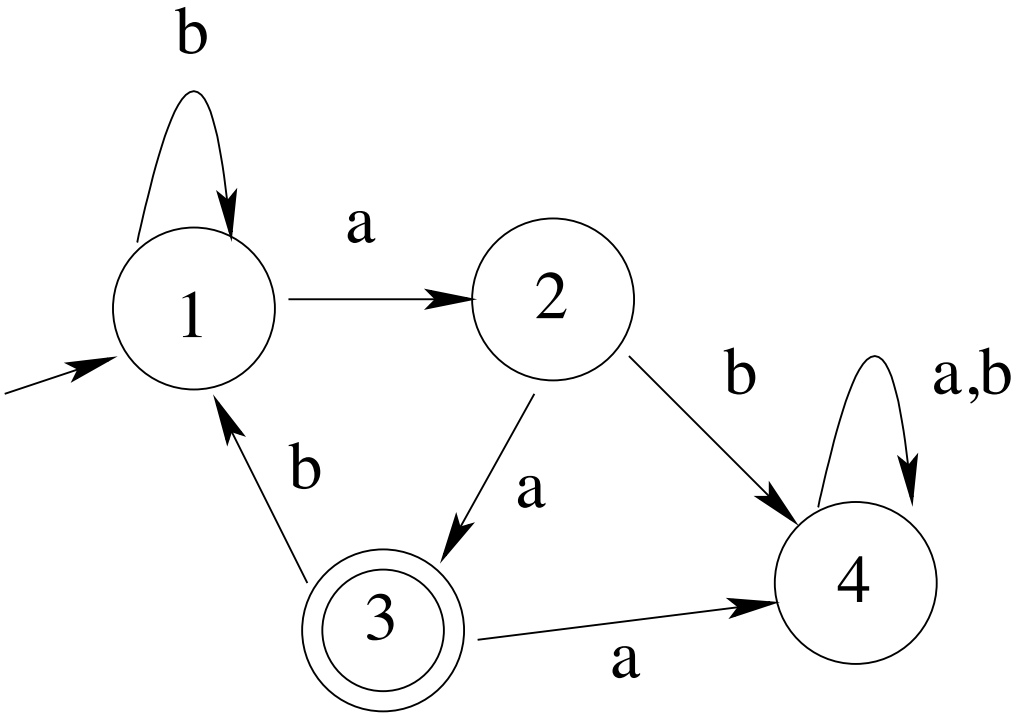
La fonction de transition itérée donne l'état final après la lecture du w par l'automate.

Exemples

$\Sigma = \{a, b\}$,
$Q = \{1, 2, 3, 4\}$,
$q_0 = 1$ est l'état initial
$A = \{3\}$ est l'ensemble des états acceptants.

T

		a	b
$\rightarrow$	1	2	1
	2	3	4
	[3]	4	1
	4	4	4



lecture de *baabbbaa*

Langage décidé

Mot accepté / refusé

Soit l'automate $M = (Q, \Sigma, T, q_0, A)$.

Un mot $w \in \Sigma^*$ est **accepté** par M ssi $T^*(q_0, w) \in A$.
Un mot $w \in \Sigma^*$ est **refusé** par M ssi $T^*(q_0, w) \notin A$.

Langage décidé

Soient M est un automate d'alphabet Σ et L un langage sur Σ

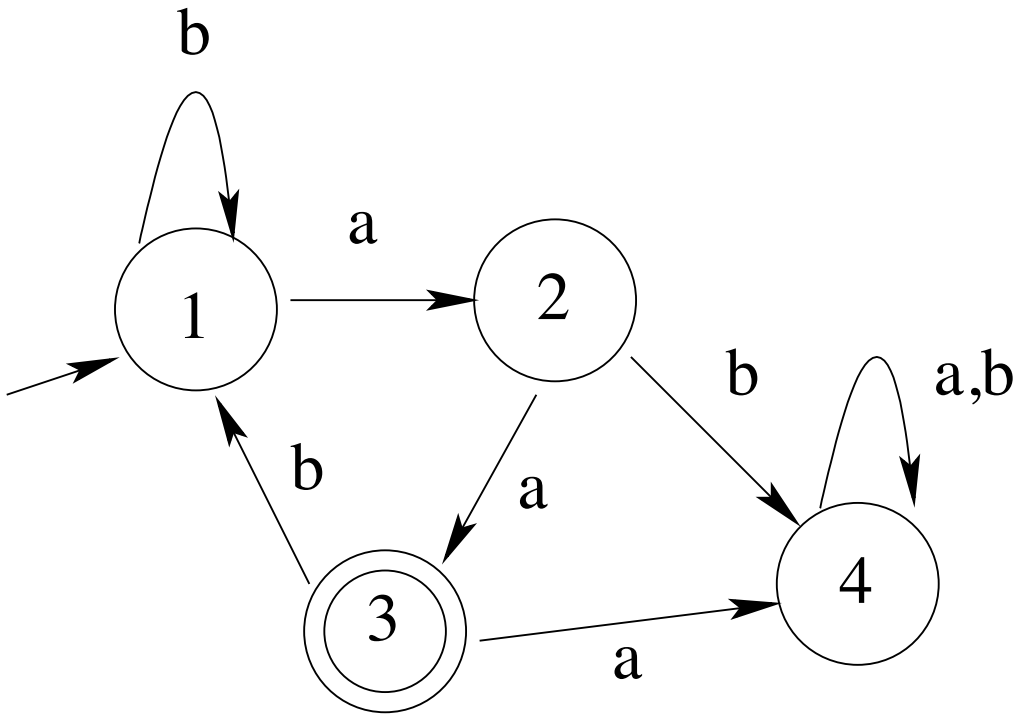
M **décide** L ssi L est l'ensemble des mots acceptés par M.

Exemples

$\Sigma = \{a, b\}$,
$Q = \{1, 2, 3, 4\}$,
$q_0 = 1$ est l'état initial
$A = \{3\}$ est l'ensemble des états acceptants.

T

		a	b
$\rightarrow$	1	2	1
	2	3	4
	$\boxed{3}$	4	1
	4	4	4



Remarque finale

A méditer

L'**état** d'un automate est la **mémoire** de la machine,

l'état de l'automate sert à mémoriser ...
...l'état de la lecture du mot (ce qui vient d'être lu)

Le nombre d'état étant fini,
il n'est possible de mémoriser qu'un nombre fini de lettres lues.