4. Les langages rationnels

4. Les langages rationnels

Hiérarchie de Chomsky :

Classe 3
$\subsetneq$ Classe 2 déterministes $\subsetneq$ Classe 2 non déterministes
$\subsetneq$ Classe 1
$\subsetneq$ Classe 0

4.1. Introduction aux langages rationnels

Classe 3 de la hiérarchie de Chomsky : **Langages rationnels** ('*regular languages*')
langages les plus simples, les plus rapides, et aussi les moins puissants.
Ils servent

- en compilation à assurer l'analyse lexicale (segmentation d'un flot de caractères en 'mots')
- pour la recherche de motifs
- pour le traitement de texte
- etc.

4.1. Introduction aux langages rationnels

Ces langages sont caractérisés de plusieurs façons : ils sont :

1. engendrés par une **grammaire régulière**.
2. décrits par une **expression régulière**.
3. engendrés par un **automate d'états finis**.

Rappel :

Définition

Une grammaire G est dite **régulière** *si toutes ses règles de production sont de la forme :*
$A \rightarrow \alpha B$, *avec :* $A \in N, \alpha \in T^*, B \in N$ *ou* $B = \epsilon$

Exemple : G_1 : $S \rightarrow aS|bS|a|b|\epsilon$

4.1. Introduction aux langages rationnels

Théorème

Un langage est **rationnel** *ssi il existe une grammaire régulière qui l'engendre.*

Exemple de grammaire réglière :

langage des mots sur $\Sigma = \{a, b\}$ qui contiennent le facteur aa :

$S \rightarrow aS|bS|aA$
$A \rightarrow aB$
$B \rightarrow aB|bB|\epsilon$

4.2. Les expressions régulières

Une expression régulière décrit un langage rationnel avec une syntaxe particulière, et correspond à une grammaire régulière.

Exemples d'expressions régulières sur $\Sigma = \{a, b\}$:

1. $(a + b)^*$ est l'ensemble des mots sur $\{a, b\}$
2. $(a + b)^* aa(a + b)^*$ langage des mots sur $\Sigma = \{a, b\}$ qui contiennent le facteur aa
3. $b^+(a + b)^*$: mots sur l'alphabet $\Sigma = \{a, b\}$ qui commencent par un ou plusieurs b

Une expression régulière est une expression algébrique qui permet de décrire un langage rationnel.

Définition

Définition inductive des expressions régulières :

- *Base :* $\emptyset$, ϵ *et les caractères de* Σ *sont des expressions régulières, représentant respectivement les langages* $\emptyset$,$\{\epsilon\}$, $\{x\}$ *si* $x \in \Sigma$.
- *Règles : si* r *et* s *sont des expressions régulières représentant les langages* R *et* S, *alors* $(r + s)$, $r.s$, r^* *et* r^+ *sont des expressions régulières représentant respectivement les langages* $R \cup S$, $R.S$, R^* *et* R^+.

En anglais : 'regular expression'

4.2. Les expressions régulières

1. $(a+b)^*$ est l'ensemble des mots sur $\{a,b\}$
2. $(a+b)^*aa(a+b)^*$ langage des mots sur $\Sigma=\{a,b\}$ qui contiennent le facteur aa
3. $b^+(a+b)^*$: mots sur l'alphabet $\Sigma=\{a,b\}$ qui commencent par un ou plusieurs b

On notera $L(r)$ le langage (rationnel) décrit par l'expression régulière r, et on dira que r **engendre** le langage $L(r)$.

Souvent, par abus de langage, on confond expression régulière et langage engendré !

Remarques :

- $r + s$ se note aussi $r|s$
- $r.s$ se note aussi rs
- $*$ a précédence sur $+$: $a + b^*$ s'interprète comme $(a + (b^*))$

Propriétés :

1. $(r^*)^* = r^*$
2. $r(r^*) = (r^*)r = r^+$
3. $(a^*b^*)^* = (a+b)^*$

4.2. Les expressions régulières

Un même langage rationnel peut être décrit par plusieurs expressions régulières différentes.

Exemple : langage des mots sur $\Sigma = \{a, b\}$ contenant le facteur 'aa' :

$r = (a + b)^* aa(a + b)^*$
$s = b^* aa(a + b)^*$

On a $L(r) = L(s)$

Définition

Si L(r)=L(s), on dira que r et s sont des expressions régulières **équivalentes**, *noté* $r \sim s$.

Théorème

Un langage est rationnel si et seulement si il existe une expression régulière le reconnaissant.

4.3. Les automates d'états finis

Les automates d'états finis (AEF)
(en anglais : "Finite State Automata" ou FA)

Ce sont les 'machines' reconnaissant les langages rationnels.

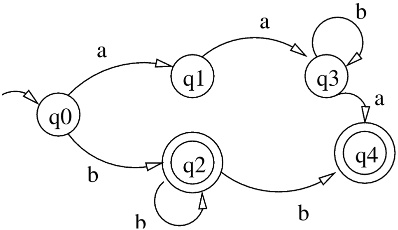
4.3. Les automates d'états finis

Un système à états finis est un modèle mathématique "discret". Il est composé d'un nombre fini de configurations, appelées des **états**, et d'actions permettant de passer d'un état à un autre. Les automates d'états finis sont des systèmes à états finis particuliers.

4.3. Les automates d'états finis

Un automate d'états finis est un graphe orienté fini dont les arcs sont étiquetés.
Il est composé :

1. d'un nombre fini de configurations (les **états**), qui sont les **sommets** du graphe
2. d'actions permettant de passer d'un état à un autre (ces actions étiquettent les **arcs** du graphe)



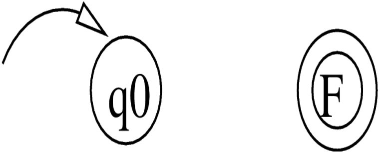
4.3. Les automates d'états finis

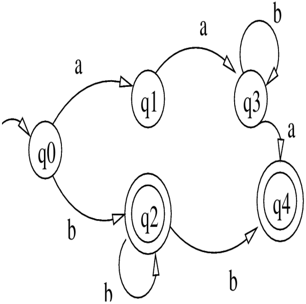
De plus, on a un (unique) **état initial** (START)
et 0, 1 ou plusieurs **états finaux** (STOP)

4.3. Les automates d'états finis

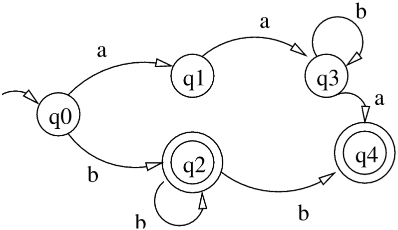
Etiquetage de l'état initial et des états finaux :

1. l'état initial est noté par une flèche
2. les états finaux ont un double cerclage

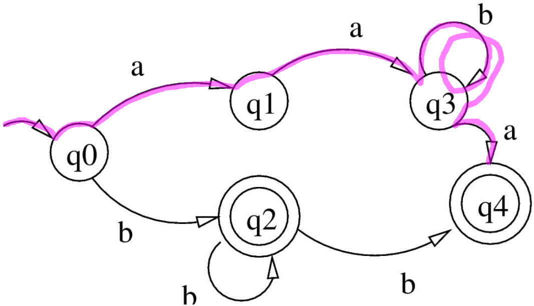




Principe (informel) :
On part de l'état initial (q_0) et on parcourt le graphe jusqu'à ce qu'on décide de s'arrêter sur un état final (ici q_2 ou q_4).



Ce parcours définit un mot du langage reconnu par l'automate.



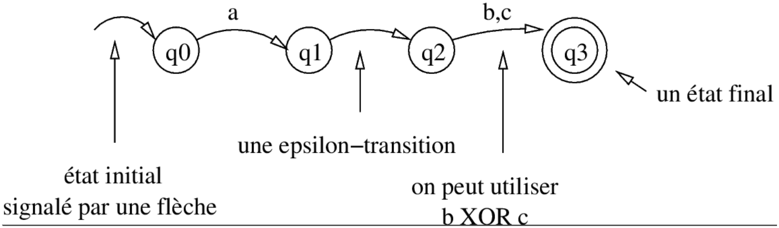
Le parcours définit le mot 'aabba'.

Définition

On peut étiqueter un arc d'un automate d'états finis par le mot vide ϵ *(souvent noté par l'absence d'un caractère),*
il correspond à une ϵ-**transition**.

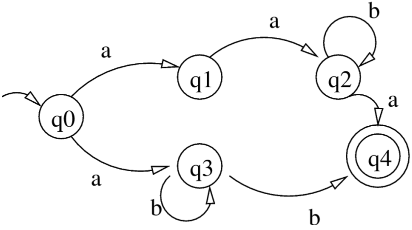
On peut étiqueter une transition par plusieurs caractères : on en choisit **un seul**.
On peut avoir des **circuits**, des **boucles de réflexivit**é. On peut avoir plusieurs arcs sortants étiquetés par un même caractère.

Exemple :



langage reconnu : $L(A) = \{ab, ac\}$

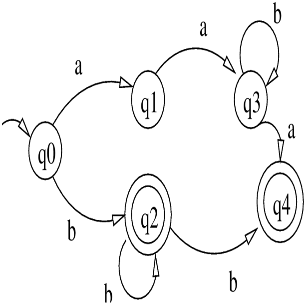
Exemple :



4.3. Les automates d'états finis

Un **automate d'états finis** est donc défini par :

- un nombre fini d'états Q (les sommets du graphe)
- un alphabet Σ.
- un ensemble fini δ de transitions (les arcs du graphe), étiquetés chacune par une (ou plusieurs) lettre(s) de Σ ou par ϵ



4.3. Les automates d'états finis

Parmi les états de Q, on distingue :

- **l'état initial** $q_0 \in Q$
  (il y a exactement un état initial)
- les **états finaux**, qui constituent l'ensemble $F \subset Q$
  (il peut y avoir plusieurs ou même aucun état final)

Formellement :

Définition

Un automate d'états finis est un quintuplet $A = (Q, \Sigma, \delta, q_0, F)$, *où :*

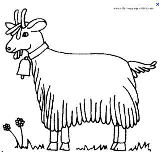
1. Q *est un* **ensemble d'états** *(de cardinal fini)*
2. Σ *est un* **alphabet** *(de cardinal fini)*
3. δ *est une* **fonction de transition** *(qui permet de passer d'un état à un autre)*
4. $q_0 \in Q$ *est* **l'état initial**
5. $F \subseteq Q$ *est l'ensemble des* **états finaux**

Exemple classique d'un automate

(extrait du livre de Hopcroft and Ullman : Introduction to Automata Theory, Languages and Computation)

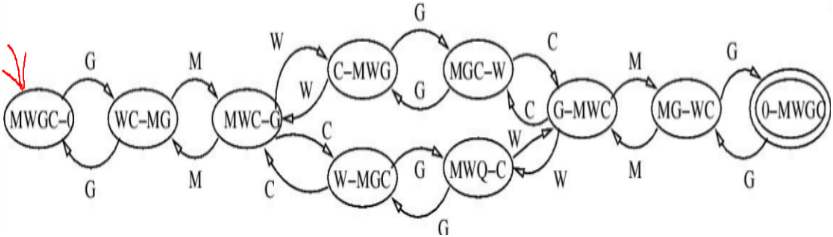
Le problème du passeur, du loup, de la chèvre et du chou.





Question :

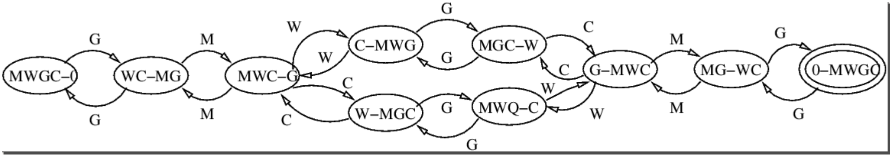
Quel est l'ensemble des solutions qui permettent au passeur
d'emmener de la rive droite à la rive gauche
le chou, la chèvre et le loup,
avec une barque ne pouvant contenir que l'un des trois,
sans laisser seuls ensemble
ni le loup et la chèvre,
ni la chèvre et le chou ?



(M - Man), le loup (W - Wolf), la chèvre (G - Goat)
et le chou (C - Cabbage).
Cet automate modélise **toutes les solutions** possibles.

On peut déduire de cette modélisation par automate d'états finis :

1. qu'il y a une solution au problème.
2. qu'il y a deux plus courtes solutions étiquetées GMWGCMG et GMCGWMG.
3. qu'il existe une infinité de solutions (le langage engendré par l'automate est infini).

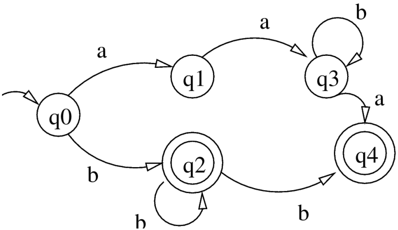


Définition

On dit qu'un automate d'états finis $A = (Q, \Sigma, \delta, q_0, F)$ **accepte** *un mot* ω *de* Σ^* *si et seulement si il existe (au moins) un chemin dans* A *allant de* q_0 *à un état final, étiqueté par les lettres successives de* ω*, entre lesquelles on a éventuellement intercalé des occurences de* ϵ*.*

Le langage $L(A)$ *reconnu par* A *est l'ensemble des mots que* A **accepte**.

Exemple 2 :

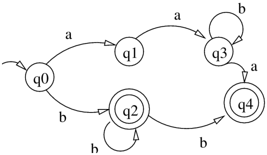


langage reconnu : $aab^*a + b^+$

On peut avoir 2 transitions possibles avec la même lettre : ici sur q_2, 2 transitions avec $'b'$.

On définit la **table de transitions** d'un automate d'états finis $A = (Q, \Sigma, \delta, q_0, F)$, qui décrit la fonction de transition δ.

Sur l'exemple 2 :



	a	b
q_0	$\{q_1\}$	$\{q_2\}$
q_1	$\{q_3\}$	$\{\emptyset\}$
q_2	$\{\emptyset\}$	$\{q_2, q_4\}$
q_3	$\{q_4\}$	$\{q_3\}$
q_4	$\{\emptyset\}$	$\{\emptyset\}$

♣ On va voir qu'il existe plusieurs sortes d'AEF.

♢ Il existe des automates plus complexes (automates à pile, machine de Turing).