Automates et Langages

IUT de Lannion

Préambule

Dans de nombreux contextes (scientifique, légal, etc.) on désigne par **langage formel** un mode d'expression plus formalisé et plus précis que le langage naturel.
C'est notamment le cas en Mathématiques et dans les langages informatiques. Dès lors l'un des enjeux est de définir précisément ce que sont les langages admissibles et comment les reconnaître.

Alphabet

Definition 1

- un **alphabet** est un ensemble fini de symboles « élémentaires », il est en général désigné par une lettre majuscule.
- Soit Σ un alphabet. Un **mot** sur Σ est une suite finie de symboles appartenant à Σ.

Exemple

- $\Sigma = \{a, b, c\}$, $\Gamma = \{0, 1\}$ sont des alphabets.
- $abbac$ et ba sont des mots sur Σ.

Alphabets

Definition 2

- la **longueur** d'un mot w est le nombre de symboles qui le composent, on la note $|w|$.
- l'unique mot de longueur 0 est le **mot vide** on le note ϵ.

Exemple

- $|abbac| = 5$
- $|ba| = 2.$

Langage

Definition 3

- L'ensemble des mots définis sur Σ est un **langage** et noté Σ^*.
- Plus généralement, on appelle **langage** défini sur Σ tout sous-ensemble L de Σ^*.

Exemple

$$\{a,b,c\}^* = \{\epsilon,a,b,c,aa,ab,ac,ba,bb,bc,ca,cb,cc,aaa,aab,\ldots\}.$$

Opérations sur les langages

Definition 4

Soient L,L_1 et L_2 trois langages définis sur un alphabet Σ, on peut construire de nouveaux langages sur Σ à l'aide :

1. des **opérations ensemblistes classiques** :
   $L_1 \cup L_2 = \{w \in \Sigma^* : \ w \in L_1 \text{ ou } w \in L_2\}$ et
   $L_1 \cap L_2 = \{w \in \Sigma^* : \ w \in L_1 \text{ et } w \in L_2\}$.

Opérations sur les langages

Definition 5

Soient L,L_1 et L_2 trois langages définis sur un alphabet Σ, on peut construire de nouveaux langages sur Σ à l'aide :

1. des **opérations ensemblistes classiques** :
   $L_1 \cup L_2 = \{w \in \Sigma^* :\ w \in L_1 \text{ ou } w \in L_2\}$ et
   $L_1 \cap L_2 = \{w \in \Sigma^* :\ w \in L_1 \text{ et } w \in L_2\}$.
2. du **produit par concaténations** : $L_1.L_2 = \{uv : u \in L_1, v \in L_2\}$.

Opérations sur les langages

Definition 6

Soient L,L_1 et L_2 trois langages définis sur un alphabet Σ, on peut construire de nouveaux langages sur Σ à l'aide :

1. des **opérations ensemblistes classiques** :
   $L_1 \cup L_2 = \{w \in \Sigma^* : \ w \in L_1 \text{ ou } w \in L_2\}$ et
   $L_1 \cap L_2 = \{w \in \Sigma^* : \ w \in L_1 \text{ et } w \in L_2\}$.
2. du **produit par concaténations** : $L_1.L_2 = \{uv : u \in L_1, v \in L_2\}$.
3. de **la puissance de** L en posant $L^0 = \{\epsilon\}$ et $L^{n+1} = L.L^n$.

Opérations sur les langages

Remarque

- On note $L^+ = \cup_{n\geq 1} L^n$
- Si $L = \emptyset$ alors $L_0 = \{\epsilon\}$.
- La **fermeture de Kleene** de L, notée L^* est définie par $L^* = \cup_{n\geq 0} L^n$

Exercice

On note $L_1 = \{0,1,2,3,4\}$ et $L_2 = \{5,6,7,8,9\}$.
Décrire les ensembles suivants :

- $L_1 \cup L_2 =$
- $L_1^2 =$
- $L_1.L_2 =$
- $(L_1 \cup L_2)^+ =$

Langages réguliers

Definition 7

Soit Σ un alphabet, on définit l'ensemble $\mathscr{R}$ des **langages réguliers** sur Σ par induction :

1. $\emptyset \in \mathscr{R}$, $\{\epsilon\} \in \mathscr{R}$, $\forall w \in \Sigma$, $\{w\} \in \mathscr{R}$.
2. Si $L_1, L_2 \in \mathscr{R}$ alors :
   - $L_1 \cup L_2 \in \mathscr{R}$.
   - $L_1 . L_2 \in \mathscr{R}$.
   - $L_1^* \in \mathscr{R}$.

Remarque

Une telle définition ensembliste permet difficilement de dire si un langage est régulier ou pas. On définit donc la notion d'**expression régulière** qui caractérise ces langages.

Langages réguliers

Definition 8

Soit Σ un alphabet. On définit l'ensemble des **expressions régulières** $\mathscr{ER}$ sur Σ par :

1. $\emptyset \in \mathscr{ER}$, $\epsilon \in \mathscr{ER}$, $\forall w \in \Sigma$, $w \in \mathscr{ER}$.
2. Si $E_1, E_2 \in \mathscr{ER}$ alors :
   - $E_1|E_2 \in \mathscr{ER}$. (E_1 ou E_2)
   - $E_1E_2 \in \mathscr{ER}$.
   - $E_1^* \in \mathscr{ER}$.

Théorème 0.1

Un langage est régulier (ou rationnel) si, et seulement si, il peut être décrit par une expression régulière.

Exemples :

Soit Σ l'alphabet $\{a,b,c\}$.

1. $(a|b|c)^*$ décrit...

Langages réguliers

Exemples :

Soit Σ l'alphabet $\{a,b,c\}$.

1. $(a|b|c)^*$ décrit...
2. $a(a|b|c)^*$ décrit...

Exemples :

Soit Σ l'alphabet $\{a,b,c\}$.

1. $(a|b|c)^*$ décrit...
2. $a(a|b|c)^*$ décrit...
3. $abc(abc)^*$ décrit...

Langages réguliers

Exemples :

Soit Σ l'alphabet $\{a,b,c\}$.

1. $(a|b|c)^*$ décrit...
2. $a(a|b|c)^*$ décrit...
3. $abc(abc)^*$ décrit...
4. $(a|b|\epsilon)c^*$ décrit...

Langages réguliers

Exemples :

Soit Σ l'alphabet $\{a,b,c\}$.

1. $(a|b|c)^*$ décrit...
2. $a(a|b|c)^*$ décrit...
3. $abc(abc)^*$ décrit...
4. $(a|b|\epsilon)c^*$ décrit...
5. $(a|\epsilon)(ba)^*(b|\epsilon)$ décrit...

Langages réguliers

Exercices :

Soit Σ l'alphabet $\{a,b,c\}$.

1. Définir les ensembles suivants : $\Sigma^*, \Sigma^+, \Sigma^3$.

Exercices :

Soit Σ l'alphabet $\{a,b,c\}$.

1. Définir les ensembles suivants : $\Sigma^*, \Sigma^+, \Sigma^3$.
2. Soient L_1 et L_2 les sous-ensembles de Σ^* définis par :

$$L_1 = a^*|b^* \; ; \; L_2 = (a|b|c)^+ a (a|b|c)^+.$$

   1. Décrire les ensembles L_1 et L_2.

Exercices :

Soit Σ l'alphabet $\{a,b,c\}$.

1. Définir les ensembles suivants : $\Sigma^*, \Sigma^+, \Sigma^3$.
2. Soient L_1 et L_2 les sous-ensembles de Σ^* définis par :

$$L_1 = a^*|b^* \ ; \ L_2 = (a|b|c)^+a(a|b|c)^+.$$

   1. Décrire les ensembles L_1 et L_2.
   2. Décrire les ensembles suivants et donner une expression régulière pour chacun d'eux : $L_1 \cup L_2$,$L_1 \cap L_2, L_1L_2, L_2L_1, L_1^*, L_2^*$.
3. Donner une expression régulière, du langage L_3 définit en compréhension par $\{a^nb^pc, n \geq 2, p \geq 0\}$.

Automates finis

Les langages réguliers étant définis, il s'agit maintenant pour nous de disposer d'un outil nous permettant de dire si oui ou non un mot donné appartient à un langage régulier.
Cet outil se formalise sous la forme d'un **automate fini**.

Automates finis

Definition 9

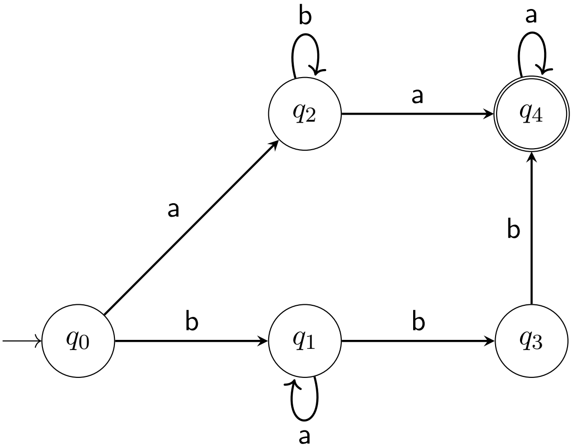
Un **Automate fini** est un quintuplet $A = (\Sigma,Q,q_0,F,\delta)$ où

1. Σ est un **alphabet (fini)**;
2. Q est un ensemble fini dont les éléments sont appelés les **états** de A;
3. $q_0 \in Q$ est l'**état initial** ;
4. $F \subset Q$ est l'ensemble des **états finaux**.
5. δ est une application d'une partie de $Q \times \Sigma$ dans Q, appelée **fonction de transition**.

Lorsque δ est définie sur $Q \times \Sigma$ tout entier, l'automate A est dit **complet**.

Il est d'usage de représenter un automate par un graphe dont les nœuds sont les états et les arêtes les couples $(q_i,q_j) \in Q^2$ étiquetés par $a \in \Sigma$ tel que $\delta(q_i,a) = q_j$.

Automates finis



Exercice

Déterminer Σ, Q, F et δ dans l'exemple ci dessus.

Automates finis

Exemple

Représenter sous forme de graphe l'automate défini par $\Sigma = \{a,b\}$; $Q = \{q_0,q_1,q_2\}$; $F = \{q_0,q_2\}$ et δ déterminée par :

δ	a	b
q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_2

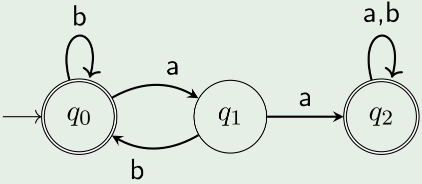
Un **chemin** dans l'automate A est une suite finie de transitions consécutives $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q_n$ débutant par l'état initial q_0. Le mot $a_1 a_2 \dots a_n$ est appelé l'**étiquette** du chemin.

Definition 10

Un chemin est dit **acceptant** lorsque l'état d'arrivée de celui-ci est un état final ; un mot de Σ^* est **reconnu** par l'automate A lorsqu'il est l'étiquette d'un chemin acceptant. Le langage $L(A)$ reconnu par l'automate est l'ensemble des mots reconnus par A.

Exemple

Donner trois mots reconnus par l'automate précédent :



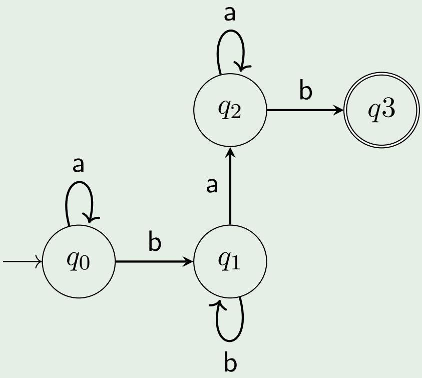
Exemples :

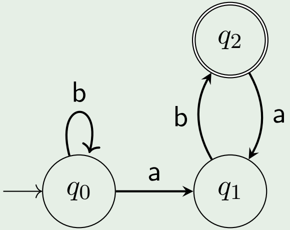
Représenter chacun des automates suivants à l'aide d'un graphe :

1. $\Sigma = \{a,b\}$; $Q = \{q_0,q_1\}$; $F = \{q_1\}$ et δ définie par : $\delta(q_0,b) = q_1$, $\delta(q_1,a) = q_1$.
2. $\Sigma = \{a,b,c\}$; $Q = \{q_0,q_1,q_2,q_3\}$; $F = \{q_2,q_3\}$ et δ définie par : $\delta(q_0,bb) = q_1$, $\delta(q_1,abb) = q_2$, $\delta(q_2,bb) = q_2$, $\delta(q_1,cc) = q_3$, $\delta(q_3,a) = q_3$

Exemples :

Soit $\Sigma = \{a,b\}$. Donner une expression régulière correspondant aux automates suivants :





Exemples

Soit un alphabet $\Sigma = \{a,b\}$.
Construire, si c'est possible, les automates reconnaissant les langages suivants :

1. $(a|b)^*$.
2. $(a|b)(a|b)^*$.
3. Tous les mots qui commencent par « a ».
4. Tous les mots qui contiennent « ab ».
5. Tous les mots qui se terminent par « ba ».
6. Tous les mots qui contiennent au plus quatre « a ».
7. Tous les mots qui contiennent un nombre pair de « b »et un nombre impair de « a ».