LASF – Mots, langages et automates

Christophe Mouilleron

école nationale supérieure d'informatique
pour l'industrie et l'entreprise

Plan

1. Mots et langages
   - Opérations sur les mots
   - Opérations sur les langages
   - Problèmes liés aux langages
2. Automates finis
   - Exemple introductif
   - Définition et représentations
   - Langage reconnu par un automate fini

Alphabet et mot

Alphabet (ou vocabulaire) = ensemble fini d'éléments appelés lettres

- l'alphabet latin $A = \{a, b, \dots, z\}$
- les chiffres décimaux $D = \{0, 1, \dots, 9\}$

Mot = suite finie de lettres $u = (u_i)_{1 \leq i \leq n} = u_1 u_2 \dots u_n$

- 2024 est un mot de 4 lettres sur D.

$$|u| = \text{longueur du mot } u$$
$$|u|_a = \text{nombre d'occurences de la lettre } a \text{ dans } u$$

Cas particulier : mot de longueur 0, appelé mot vide et noté ε

Notion de langage

Soit A un alphabet fini. On note :

A^* = ensemble des mots sur A

A^+ = ensemble des mots **non vides** sur A $\qquad A^* = \{\varepsilon\} \sqcup A^+$

Langage : sous-ensemble L de A^* $\qquad L \in \mathfrak{P}(A^*),\ L \subseteq A^*$

- ensemble des mots français (langage sur l'alphabet latin)
- ensemble des écritures des entiers (langage sur D)

On parle de mot/langage « sur A » ou « de A^* ».

Plan

1. Mots et langages
   - Opérations sur les mots
   - Opérations sur les langages
   - Problèmes liés aux langages

2. Automates finis
   - Exemple introductif
   - Définition et représentations
   - Langage reconnu par un automate fini

Concaténation de deux mots

Loi de composition interne pour A^*, notée $\cdot$

Définition

Si $u = u_1 \dots u_m$ et $v = v_1 \dots v_n$, alors $w = u \cdot v := w_1 \dots w_{m+n}$ avec :

$$\begin{aligned} w_i &:= u_i && \text{pour } 1 \le i \le m, \\ w_i &:= v_{i-m} && \text{pour } m+1 \le i \le m+n. \end{aligned}$$

Exemple : $123 \cdot 456 = 123456$

Propriétés :

Concaténation de deux mots

Loi de composition interne pour A^*, notée $\cdot$

Définition

Si $u = u_1 \dots u_m$ et $v = v_1 \dots v_n$, alors $w = u \cdot v := w_1 \dots w_{m+n}$ avec :

$$\begin{aligned} w_i &:= u_i && \text{pour } 1 \leq i \leq m, \\ w_i &:= v_{i-m} && \text{pour } m+1 \leq i \leq m+n. \end{aligned}$$

Exemple : $123 \cdot 456 = 123456$

Propriétés :

- Élement neutre : $u \cdot \varepsilon = \varepsilon \cdot u = u$
- Associativité : $(u \cdot v) \cdot w = u \cdot (v \cdot w)$

$\Rightarrow (A^*, \cdot)$ est un monoïde

- $|u \cdot v| = |u| + |v| \qquad |u \cdot v|_a = |u|_a + |v|_a$

Concaténation de deux mots

Loi de composition interne pour A^*, notée $\cdot$ (ou pas)

Définition

Si $u = u_1 \dots u_m$ et $v = v_1 \dots v_n$, alors $w = uv := w_1 \dots w_{m+n}$ avec :

$$\begin{aligned} w_i &:= u_i && \text{pour } 1 \leq i \leq m, \\ w_i &:= v_{i-m} && \text{pour } m+1 \leq i \leq m+n. \end{aligned}$$

Exemple : $123 \cdot 456 = 123456$

Propriétés :

- Élement neutre : $u \cdot \varepsilon = \varepsilon \cdot u = u$
- Associativité : $(uv)w = u(vw)$
- $|uv| = |u| + |v| \qquad |uv|_a = |u|_a + |v|_a$

$\Rightarrow (A^*, \cdot)$ est un monoïde

Puissance d'un mot

Soit $u \in A^*$.

Définition

Les puissances de u sont définies récursivement par :

$$\begin{cases} u^0 & := \varepsilon \\ u^n & := u \cdot u^{n-1} \end{cases}$$

Cas particulier : le carré de u est $u^2 = uu$

Propriétés :

- $|u^n| = n\,|u|$
- $|u^n|_a = n\,|u|_a$
- $\varepsilon^n = \varepsilon$

Préfixes d'un mot

Définition

$v \in A^*$ est un préfixe de $u \in A^*$ si il existe $w \in A^*$ tel que $u = vw$.

On note $\text{Pré}(u)$ l'ensemble des préfixes de u.
Si $w \neq \varepsilon$, on parle de préfixe propre.

Propriétés :

- $\text{Pré}(\varepsilon) = \{\varepsilon\}$
- $\varepsilon \in \text{Pré}(u)$ vrai quelque soit u
- $|v| \leq |u|$ ($<$ si prefixe propre)
- $|v|_a \leq |u|_a$

Suffixes et facteurs d'un mot

Définition

$v \in A^*$ est un suffixe de $u \in A^*$ si il existe $w \in A^*$ tel que $u = wv$.

On note $\mathrm{Suf}(u)$ l'ensemble des suffixes de u.
Si $w \neq \varepsilon$, on parle de suffixe propre.

Définition

$v \in A^*$ est un facteur de $u \in A^*$ si il existe $w_1 \in A^*$ et $w_2 \in A^*$ tels que $u = w_1 v w_2$.

On note $\mathrm{Fact}(u)$ l'ensemble des facteurs de u.
Si $w_1 w_2 \neq \varepsilon$, on parle de facteur propre.

Résiduels d'un mot

Calcul de résiduel : opération « inverse » de la concaténation.

Définition

Si $v \in \text{Pré}(u)$, le résiduel (à gauche) de u par v est le mot $w \in A^*$ tel que $u = vw$. On note ausi ce mot $v^{-1}u$.

Définition

Si $v \in \text{Suf}(u)$, le résiduel (à droite) de u par v est le mot $w \in A^*$ tel que $u = wv$. On note aussi ce mot uv^{-1}.

Exemples :

- $(202)^{-1} \cdot 2024 = 4$ et $2024 \cdot (24)^{-1} = 20$
- $3^{-1} \cdot 2024 = ???$ résultat non défini

Transposé d'un mot

Soit $u = u_1 \dots u_n \in A^*$.

Définition

Le transposé de u est le mot $u^t := u_n u_{n-1} \dots u_1$.

Exemples :

- $2024^t = 4202$
- $\texttt{radar}^t = \texttt{radar}$

`radar` est un palindrome

Propriétés :

Transposé d'un mot

Soit $u = u_1 \dots u_n \in A^*$.

Définition

Le transposé de u est le mot $u^t := u_n u_{n-1} \dots u_1$.

Exemples :

- $2024^t = 4202$
- $\texttt{radar}^t = \texttt{radar}$ — `radar` est un palindrome

Propriétés :

- $(u^t)^t = u$ et $(uv)^t = v^t u^t$
- $|u^t| = |u|$ et $|u^t|_a = |u|_a$
- $v \in \text{Pré}(u^t) \Leftrightarrow v^t \in \text{Suf}(u)$
- $v \in \text{Fact}(u^t) \Leftrightarrow v^t \in \text{Fact}(u)$

Plan

Opérations ensemblistes

Soient $L_1, L_2 \subseteq A^*$.

Union : $L_1 \cup L_2 := \{u \in A^*,\ u \in L_1 \text{ ou } u \in L_2\}$

Intersection : $L_1 \cap L_2 := \{u \in A^*,\ u \in L_1 \text{ et } u \in L_2\}$

Complémentaire : $L_1^C := \{u \in A^*,\ u \notin L_1\}$

Différence : $L_1 \setminus L_2 := \{u \in A^*,\ u \in L_1 \text{ et } u \notin L_2\}$

Différence symétrique : $L_1 \Delta L_2 := L_1 \setminus L_2 \cup L_2 \setminus L_1 = (L_1 \cup L_2) \setminus (L_1 \cap L_2)$

Produit de deux langages

Produit : étend la concaténation des mots aux langages

Définition

Le produit de deux langages L_1 et L_2 est le langage
$L_1 \cdot L_2 := \{u \cdot v,\ u \in L_1 \text{ et } v \in L_2\}$.

Produit de deux langages

Produit : étend la concaténation des mots aux langages

Définition

Le produit de deux langages L_1 et L_2 est le langage
$L_1 \cdot L_2 := \{u \cdot v,\ u \in L_1 \text{ et } v \in L_2\}$.

Exemple :

Si

- $L_1 = \{a^n,\ n \in \mathbb{N}^*\}$ = mots non vides qui ne contiennent que la lettre a
- $L_2 = \{b^m,\ m \in \mathbb{N}^*\}$ = mots non vides qui ne contiennent que la lettre b

Alors :

- $L_1 \cdot L_1 = \{a^n,\ n \geq 2\}$ = ensemble des mots de taille ≥ 2 avec que des a
- $L_1 \cdot L_2 = \{a^n b^m,\ n \in \mathbb{N}^* \text{ et } m \in \mathbb{N}^*\}$

Puissances et étoile d'un langage

Définition

Soit $L \subseteq A^*$. Les puissances de L sont définies récursivement par :

$$\begin{cases} L^0 & := \{\varepsilon\} \\ L^n & := L \cdot L^{n-1} \end{cases}$$

Étoile d'un langage : $L^* := \bigcup_{k \geq 0} L^k$

Autre notation : $L^+ := \bigcup_{k > 0} L^k$

Puissances et étoile d'un langage

Définition

Soit $L \subseteq A^*$. Les puissances de L sont définies récursivement par :

$$\begin{cases} L^0 & := \{\varepsilon\} \\ L^n & := L \cdot L^{n-1} \end{cases}$$

Étoile d'un langage : $L^* := \bigcup_{k \geq 0} L^k$

Autre notation : $L^+ := \bigcup_{k > 0} L^k$

Attention : Ne pas confondre L^n et $\{u^n,\ u \in L\}$!

Extension des autres opérations sur les mots

Préfixe : $\text{Pré}(L) := \bigcup_{u \in L} \text{Pré}(u)$
Suffixe : $\text{Suf}(L) := \bigcup_{u \in L} \text{Suf}(u)$
Facteur : $\text{Fact}(L) := \bigcup_{u \in L} \text{Fact}(u)$ $= \text{Pré}\,(\text{Suf}(L))$

Transposé : $L^t := \{u^t,\ u \in L\}$

Résiduels (à gauche) par u : $u^{-1}L := \{v \in A^*,\ uv \in L\}$
Résiduels (à droite) par u : $Lu^{-1} := \{v \in A^*,\ vu \in L\}$

Plan

1. Mots et langages
   - Opérations sur les mots
   - Opérations sur les langages
   - Problèmes liés aux langages
2. Automates finis
   - Exemple introductif
   - Définition et représentations
   - Langage reconnu par un automate fini

Problèmes liés aux langages

Représentation des langages

Déterminer efficacement si :

- L est vide, fini ou infini — finitude
- $u \in A^*$ appartient à L — reconnaissance
- deux représentations correspondent au même langage — égalité

Génération des mots u de L

Problèmes liés aux langages rationnels

Représentation des langages
$\rightsquigarrow$ expressions rationnelles

Déterminer efficacement si :

- L est vide, fini ou infini — finitude
- $u \in A^*$ appartient à L — reconnaissance
- deux représentations correspondent au même langage — égalité
- L est rationnel

$\rightsquigarrow$ automates finis

Génération des mots u de L
$\rightsquigarrow$ grammaires linéaires gauches et droites

Plan

Calcul du reste d'une division

Objectif : Calculer le reste de n/d en utilisant une mémoire finie

Entrée du problème : $n \in \mathbb{N}$ d est fixé

Problème traité par Blaise Pascal (1623 – 1662) dans son `De Numeribus Multiplicibus`

Cas faciles :

- $d = 2$ et $d = 5$ selon dernier chiffre
- $d = 3$ et $d = 9$ selon somme des chiffres
- $d = 11$ selon somme alternée des chiffres

Cas général : le ruban de Pascal

Notons $n = \sum_{i \geq 0} n_i \cdot 10^i$.

Pour $d = 11$:

i	0	1	2	3	...	8	9	...
$10^i \bmod 11$	1	-1	1	-1	...	1	-1	...

$\rightsquigarrow n \equiv \sum_{i \geq 0} n_i \cdot (-1)^i \bmod 11$.

Cas général :

1. calcul du tableau (le ruban) de proche en proche
2. calcul de la somme modulo d
3. réitérer tant que résultat $\geq \max\{d, 10\}$

Exemple pour $n = 2026$ et $d = 7$

1. calcul du tableau :

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
$10^i \bmod 7$	1	3													...

Exemple pour $n = 2026$ et $d = 7$

1. calcul du tableau :

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
$10^i \bmod 7$	1	3	2												...

Exemple pour $n = 2026$ et $d = 7$

1. calcul du tableau :

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
$10^i \bmod 7$	1	3	2	6											...

Exemple pour $n = 2026$ et $d = 7$

1. calcul du tableau :

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
$10^i \bmod 7$	1	3	2	6	4										...

Exemple pour $n = 2026$ et $d = 7$

1. calcul du tableau :

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
$10^i \bmod 7$	1	3	2	6	4	5	1	3	2	6	4	5	1	3	...

2. calcul de la somme modulo 7 :

$$6 \times 1 + 2 \times 3 + 0 \times 2 + 2 \times 6 = 24$$

Exemple pour $n = 2026$ et $d = 7$

1. calcul du tableau :

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
$10^i \bmod 7$	1	3	2	6	4	5	1	3	2	6	4	5	1	3	...

2. calcul de la somme modulo 7 :

$$6 \times 1 + 2 \times 3 + 0 \times 2 + 2 \times 6 = 24$$

3. itérations :

$$24 \rightsquigarrow 4 \times 1 + 2 \times 3 = 10$$
$$10 \rightsquigarrow 0 \times 1 + 1 \times 3 = 3$$

La solution de Pascal (suite)

Mémoire finie ? Non.

- Le tableau possède a priori un nombre infini de colonnes.

La solution de Pascal (suite)

Mémoire finie ? Non.

- Le tableau possède a priori un nombre infini de colonnes.

Propriété

La suite $(10^n \bmod d)_{n \in \mathbb{N}}$ est périodique de période au plus $d - 1$ à partir d'un certain rang.

$\rightsquigarrow$ stockage d'un nombre fini de colonnes

La solution de Pascal (suite)

Mémoire finie ? Non.

- Le tableau possède a priori un nombre infini de colonnes.

Propriété

La suite $(10^n \bmod d)_{n\in\mathbb{N}}$ est périodique de période au plus $d-1$ à partir d'un certain rang.

$\rightsquigarrow$ stockage d'un nombre fini de colonnes

- La somme accumulée peut être arbitrairement grande.

La solution de Pascal (suite)

Mémoire finie ? Non.

- Le tableau possède a priori un nombre infini de colonnes.

Propriété

La suite $(10^n \bmod d)_{n\in\mathbb{N}}$ est périodique de période au plus $d-1$ à partir d'un certain rang.

$\rightsquigarrow$ stockage d'un nombre fini de colonnes

- La somme accumulée peut être arbitrairement grande.

$\rightsquigarrow$ faire une accumulation modulo d

Vers une machine finie

En fait, pour effectuer le calcul de $n \bmod d$, on a besoin de connaître :

- la valeur dans le ruban $\leq d-1$ positions
- la valeur accumulée jusque là $0 \leq \star \leq d-1$

$\rightsquigarrow \leq (d-1) \times d = d(d-1)$ états

Une étape de calcul =

1. lecture d'un chiffre
2. mise à jour de la valeur accumulée

$\rightsquigarrow$ transition vers un nouvel état selon l'état courant et le chiffre lu

Machine donnant le reste par 3 d'un nombre en binaire

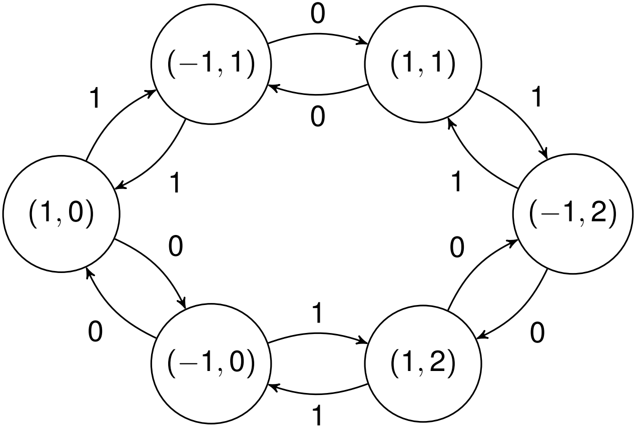
i	0	1
$2^i \bmod 3$	1	-1

$\rightsquigarrow 2 \times 3 = 6$ états

Machine donnant le reste par 3 d'un nombre en binaire

i	0	1
$2^i \bmod 3$	1	-1

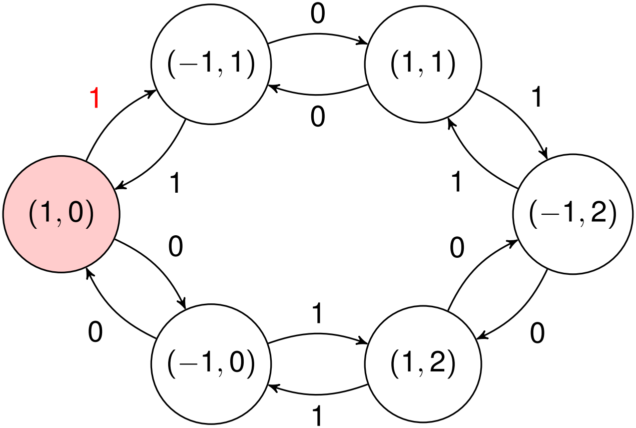
$\rightsquigarrow 2 \times 3 = 6$ états



Machine donnant le reste par 3 d'un nombre en binaire

i	0	1
$2^i \bmod 3$	1	-1

$\rightsquigarrow 2 \times 3 = 6$ états



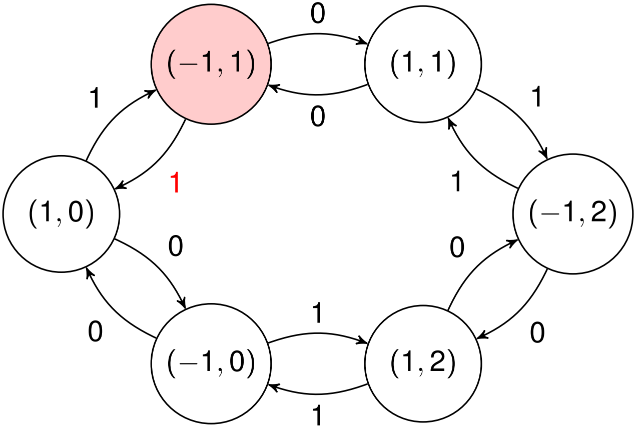
Exemple :

$$n = 23$$
$$= \overline{10111}^2$$

Machine donnant le reste par 3 d'un nombre en binaire

i	0	1
$2^i \bmod 3$	1	-1

$\rightsquigarrow 2 \times 3 = 6$ états



Exemple :

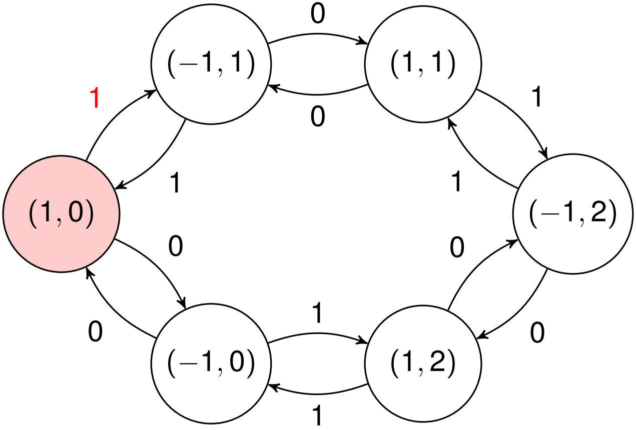
$n = 23$

$= \overline{10111}^2$

Machine donnant le reste par 3 d'un nombre en binaire

i	0	1
$2^i \bmod 3$	1	-1

$\rightsquigarrow 2 \times 3 = 6$ états



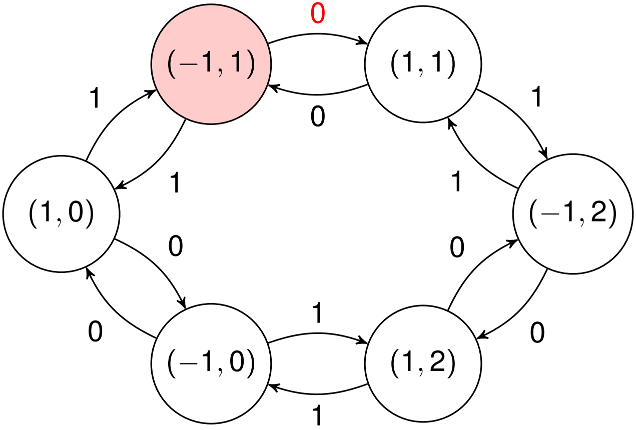
Exemple :

$$n = 23$$
$$= \overline{10111}^{2}$$

Machine donnant le reste par 3 d'un nombre en binaire

i	0	1
$2^i \bmod 3$	1	-1

$\rightsquigarrow 2 \times 3 = 6$ états



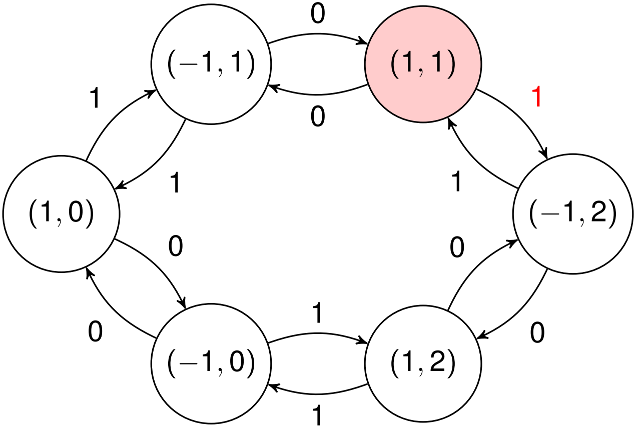
Exemple :

$$n = 23$$
$$= \overline{10111}^2$$

Machine donnant le reste par 3 d'un nombre en binaire

i	0	1
$2^i \bmod 3$	1	-1

$\rightsquigarrow 2 \times 3 = 6$ états



Exemple :

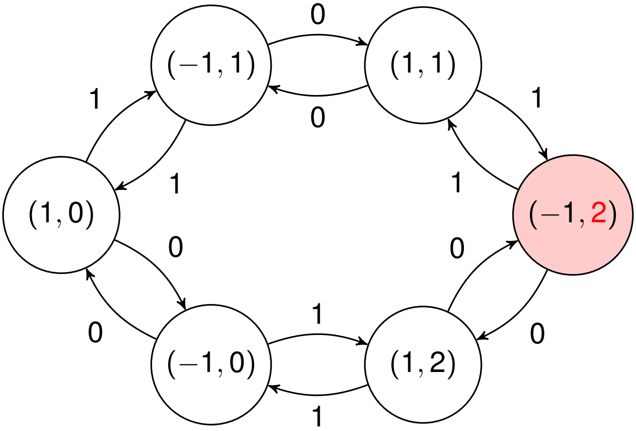
$n = 23$

$= \overline{10111}^2$

Machine donnant le reste par 3 d'un nombre en binaire

i	0	1
$2^i \bmod 3$	1	-1

$\rightsquigarrow 2 \times 3 = 6$ états



Exemple :

$$n = 23$$
$$= \overline{10111}^2$$

$23 \bmod 3 = 2$

Plan

Définition

Un automate fini est un quintuplet $\mathcal{A} := (A, Q, I, F, E)$ où :

- A est l'alphabet $\{0, 1\}$
- Q est un ensemble fini d'états $\{(i, j) \mid i = \pm 1,\ 0 \leq j \leq 2\}$
- $I \subseteq Q$ est l'ensemble des états initiaux $\{(1, 0)\}$
- $F \subseteq Q$ est l'ensemble des états finaux $\{(1, 0),\ (-1, 0)\}$
- $E \subseteq Q \times A \times Q$ est un ensemble de transitions

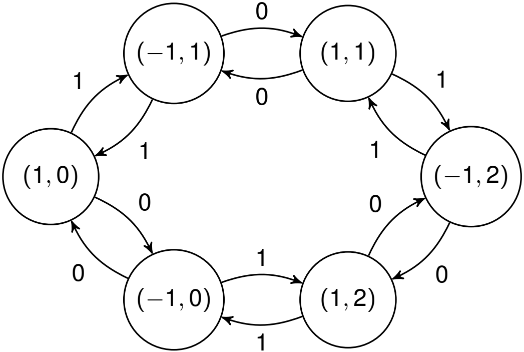
$$\Big((i, j),\ c,\ (i', j')\Big) \in E \text{ si et seulement si } \begin{cases} i' & = -i \\ j' & = j + i \cdot c \bmod 3. \end{cases}$$

État $q \in Q$: q

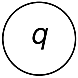
Transition $(p, a, q) \in E$:

$\xrightarrow{a}$ de p à q

Sur l'exemple précédent :



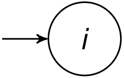
Représentation graphique

État $q \in Q$: 

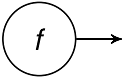
Transition $(p, a, q) \in E$:

$\xrightarrow{a}$ de p à q

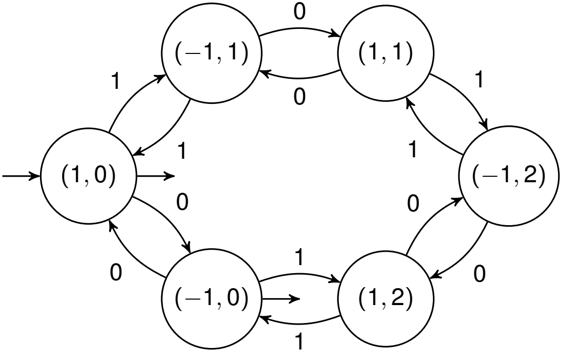
État initial $i \in I$:



État final $f \in F$:

ou


Sur l'exemple précédent :



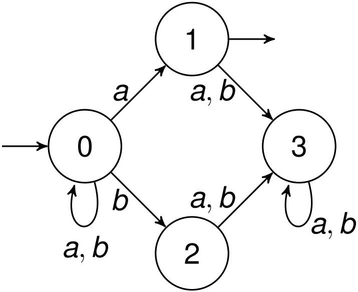
Plan

1. Mots et langages
   - Opérations sur les mots
   - Opérations sur les langages
   - Problèmes liés aux langages
2. Automates finis
   - Exemple introductif
   - Définition et représentations
   - Langage reconnu par un automate fini

Chemin dans un automate

Chemin dans un automate : suite de transitions de E

Exemples :



- $0 \xrightarrow{a} 0 \xrightarrow{a} 0 \xrightarrow{a} 1$
- $0 \xrightarrow{b} 2 \xrightarrow{a} 3 \xrightarrow{b} 3 \xrightarrow{a} 3$
- $0 \xrightarrow{b} 0 \xrightarrow{a} 0 \xrightarrow{b} 0 \xrightarrow{a} 0$

Reconnaissance d'un mot par un automate fini

Définition

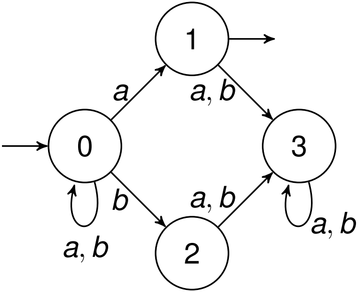
Si $p = s_0 \xrightarrow{u_1} s_1 \xrightarrow{u_2} s_2 \dots s_{n-1} \xrightarrow{u_n} s_n = q$ est un chemin dans l'automate $\mathcal{A} = (A, Q, I, F, E)$, on dit que le mot $u = u_1 \dots u_n$ est l'étiquette de ce chemin.

Notation : $p \xrightarrow{u} q$ peut correspondre à plusieurs chemins

Définition

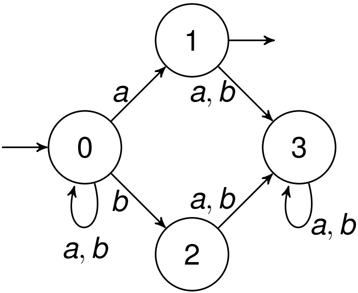
On dit qu'un mot $u \in A^*$ est reconnu (ou accepté) par l'automate fini $\mathcal{A} = (A, Q, I, F, E)$ lorsqu'il existe deux états $i \in I$ et $f \in F$, et un chemin dans l'automate de i à f étiqueté par u.

Exemple de mot reconnu



Chemins étiquetés par *baba* :

Exemple de mot reconnu

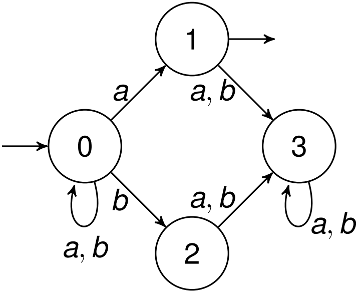


Chemins étiquetés par *baba* :

- $2 \xrightarrow{b} 3 \xrightarrow{a} 3 \xrightarrow{b} 3 \xrightarrow{a} 3$ NON
- $0 \xrightarrow{b} 0 \xrightarrow{a} 0 \xrightarrow{b} 0 \xrightarrow{a} 0$ NON
- $0 \xrightarrow{b} 0 \xrightarrow{a} 0 \xrightarrow{b} 0 \xrightarrow{a} 1$ OUI
- ... peu importe

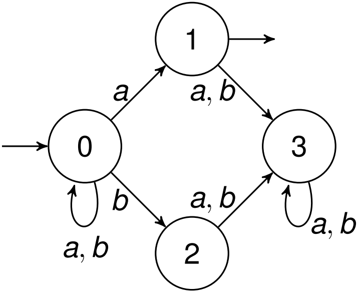
$\Rightarrow$ *baba* est reconnu par $\mathcal{A}$.

Exemple de mot non reconnu



Chemins étiquetés par *bbb* :

Exemple de mot non reconnu



Chemins étiquetés par *bbb* :

- $0 \xrightarrow{b} 0 \xrightarrow{b} 0 \xrightarrow{b} 0$ NON
- $0 \xrightarrow{b} 0 \xrightarrow{b} 0 \xrightarrow{b} 2$ NON
- $0 \xrightarrow{b} 0 \xrightarrow{b} 2 \xrightarrow{b} 3$ NON
- $0 \xrightarrow{b} 2 \xrightarrow{b} 3 \xrightarrow{b} 3$ NON
- ... (ne commence pas par 0) NON

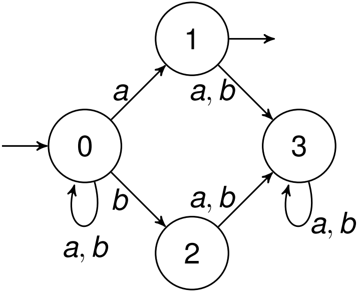
$\Rightarrow$ *bbb* n'est pas reconnu par $\mathcal{A}$.

Langage reconnu par un automate fini

Définition

Le langage reconnu par l'automate fini $\mathcal{A} = (A, Q, I, F, E)$ est l'ensemble des mots reconnus par $\mathcal{A}$. On le note $\mathrm{L}(\mathcal{A})$.

Exemple :

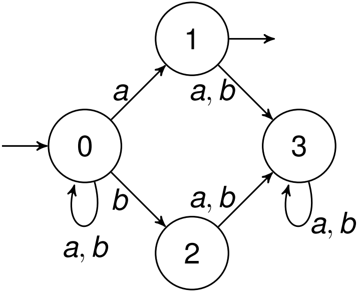


Langage reconnu par un automate fini

Définition

Le langage reconnu par l'automate fini $\mathcal{A} = (A, Q, I, F, E)$ est l'ensemble des mots reconnus par $\mathcal{A}$. On le note $\mathrm{L}(\mathcal{A})$.

Exemple :



- mots se terminant par a reconnus
- mots se terminant par b non reconnus
- mot vide non reconnu

$$\Rightarrow \mathrm{L}(\mathcal{A}) = \{ua,\ u \in A^*\}$$

Langages reconnaissables

Définition

Un langage $L \subset A^*$ pour lequel il existe un automate fini $\mathcal{A} = (A, Q, I, F, E)$ tel que $\mathrm{L}(\mathcal{A}) = L$ est dit reconnaissable (sous entendu, par un automate fini).

Notation : $\mathrm{Rec}(A^*)$ est l'ensemble des langages reconnaissables.

Propriétés :

- $\emptyset \in \mathrm{Rec}(A^*)$
- $\{u\} \in \mathrm{Rec}(A^*)$ pour tout $u \in A^*$
- $A^* \in \mathrm{Rec}(A^*)$

Exemples de langages reconnus par un automate fini

Langage vide :

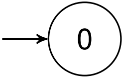
Langage $\{\varepsilon\}$:

Langage réduit à un mot :

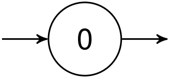
Langage A^* :

Exemples de langages reconnus par un automate fini

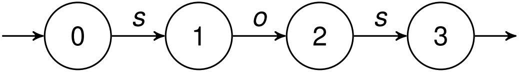
Langage vide :



Langage $\{\varepsilon\}$:



Langage réduit à un mot :



Langage A^* :