IN 406 – Théorie des Langages
Cours 5 : Langage non-régulier, lemme de l'étoile

Franck Quessette – Franck.Quessette@uvsq.fr

Université de Versailles – Saint-Quentin

V4 2020–2021

Rappel 1

Notions déjà vues :

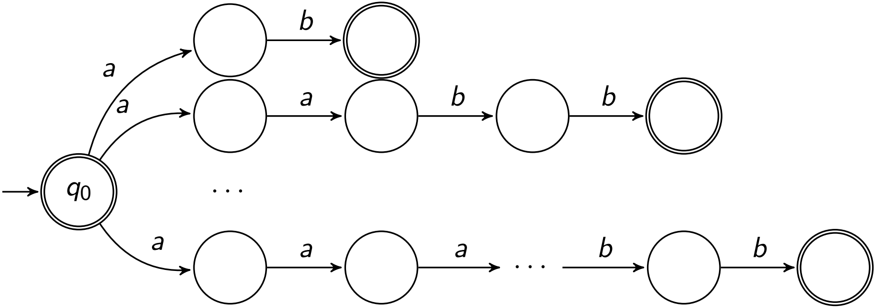
- lettre, alphabet $\boxed{\textbf{fini}}$;
- mot, langage $\boxed{\textbf{fini}}$ ou $\boxed{\textbf{infini}}$;
- langage rationnel (opérations ensemblistes) ;
- automate $\boxed{\textbf{fini}}$ non-déterminsite (AFN), automate $\boxed{\textbf{fini}}$ déterministe ;
- langage reconnaissable par automate $\boxed{\textbf{fini}}$;
- expression régulière.
- langage reconnaissable par une expression régulière.

Langage non reconnaissable par un automate fini 1

Langage fini

Soit $L = \{a^n b^n, 0 \leq n \leq K\}$, est fini, il contient $K + 1$ mots.

Automate



Langage non reconnaissable par un automate fini 1

Théorème

Soit $L = \{a^n b^n, 0 \leq n\}$, il n'existe pas d'automate fini reconnaissant exactement L.

Preuve

Par l'absurde : supposons qu'un automate sans ε-transition $\mathcal{A}$ reconnaisse exactement L. Soit N le nombre d'états de cet automate et considérons le mot $w = a^N b^N$.
Soient $q_0^{(w)}, q_1^{(w)}, \ldots, q_{2\times N}^{(w)}$ ($q_0^{(w)} = q_0$ et $q_{2\times N}^{(w)} \in F$) la suite des états utilisés lors de la reconnaissance de w. w étant plus long que le nombre d'états de l'automate, la suite des états contient nécessairement un état qui apparait au moins deux fois. Soit $\boxed{q}$ le premier état dans la suite des $q_i^{(w)}$ qui apparait deux fois.

Langage non reconnaissable par un automate fini 2

Preuve (suite)

La reconnaissance de w peut être factorisée en trois morceaux :

- De l'état initial $q_0^{(w)}$ jusqu'au premier passage par l'état $\boxed{q}$, ce qui reconnait un mot w_1.
- Du premier passage par $\boxed{q}$ jusqu'au deuxième passage par $\boxed{q}$, ce qui reconnait un mot w_2.
- Du deuxième passage par $\boxed{q}$ jusqu'à l'état final $q_{2\times N}^{(w)}$, ce qui reconnait un mot w_3.

Donc $w = w_1 w_2 w_3$ avec $0 < |w_2| \leq N$. Le mot $w' = w_1 w_2 w_2 w_3$ est également reconnu par le langage puisque pour reconnaitre w_2 on commence par $\boxed{q}$ et on termine par $\boxed{q}$. Ce cycle peut donc être emprunté autant de fois que l'on veut.

Langage non reconnaissable par un automate fini 3

Preuve (suite)

On va montrer que $w' \notin L$, ce qui est contradictoire puisque l'automate est censé reconnaître exactement L.
Comme $w = w_1 w_2 w_3 = a^N b^N$, il y a trois "découpages" possibles de w :

- w_2 ne contient que des a : $w_1 = a^x$, $w_2 = a^y$ $(y > 0)$ et $w_3 = a^{N-x-y} b^N$. Dans ce cas $w' = a^{N+y} b^N$ et $w' \notin L$.
- w_2 contient des a et des b : $w_1 = a^{N-x}$, $w_2 = a^x b^y$ $(x + y > 0)$ et $w_3 = b^{N-y}$. Dans ce cas $w' = a^N b^y a^x b^N$ et $w' \notin L$.
- w_2 ne contient que des b : $w_1 = a^N b^x$, $w_2 = b^y$ $(y > 0)$ et $w_3 = b^{N-x-y}$. Dans ce cas $w' = a^N b^{N+y}$ et $w' \notin L$.

On a donc une contradiction ce qui prouve qu'un automate fini reconnaissant L n'existe pas

Pourquoi $a^n b^n$ pose problème ?

Idée intuitive

- Dans le langage $a^n b^n$, il faut "compter" le nombre de a pour faire le même nombre de b. Ce compteur n'est pas borné.
- L'automate n'a comme mémoire uniquement l'état courant, comme le nombre d'états est fini, on ne peut pas mémoriser un nombre quelconque de valeurs.
- On va, par la suite rajouter une pile pour pouvoir faire compteur infini.

Lemme de l'étoile (ou lemme de la pompe)

Lemme de l'étoile

SI L est reconnaissable par un automate fini $\mathcal{A}$ à N états,

ALORS

$\forall w \in L$ avec $|w| \geq N$, $\exists$ factorisation $w = xyz$ avec $|y| > 0$ telle que $\forall n \geq 0$, xy^nz est reconnu par $\mathcal{A}$ et donc appartient à L.

Utilisation du lemme de l'étoile

Utilisation de la **contraposée** du lemme pour montrer qu'un langage n'est pas régulier :

SI $\exists w \in L, \forall x, y, z$ avec $w = xyz$ et $|y| > 0$, $\exists n > 1, xy^nz \notin L$

ALORS

L n'est pas régulier

Preuve du lemme de l'étoile

Preuve

Même idée que pour $a^n b^n$.
S'il n'existe pas de mot plus long que $|Q|$, le lemme est vérifié.
Soit w, tel que $|W| > |Q|$. Soient $q_0^{(w)}, q_1^{(w)}, \ldots, q_{|w|}^{(w)}$ ($q_0^{(w)} = q_0$ et $q_{|w|}^{(w)} \in F$) la suite des états utilisés lors de la reconnaissance de w.
w étant plus long que le nombre d'états de l'automate, la suite des états contient nécessairement un état qui apparait au moins deux fois. Soit $\boxed{q}$ le premier état dans la suite des $q_i^{(w)}$ qui apparait deux fois.

Preuve du lemme de l'étoile

Preuve

w peut être factorisé en xyz avec :

- x le mot reconnu de $q_0^{(w)}$ au premier passage par $\boxed{q}$;
- y le mot reconnu entre les deux premiers passages par $\boxed{q}$,
- $0 < |y|$ car il n'y a pas d'ε-transition ;
- $|y| \leq |Q|$ car $\boxed{q}$ est le premier état rencontré deux fois ;
- z le mot reconnu entre $\boxed{q}$ et l'état final $q_{|w|}^{(w)}$.

On peut donc reconnaitre plusieurs y consécutifs et donc $xy^nz \in L$ pour tout entier $n \geq 0$

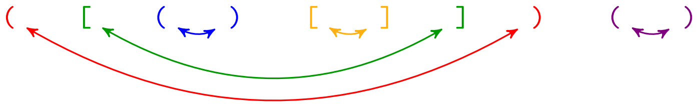
Exemples

- le langage des mots ayant autant de a que de b ;
- le langage des palindromes ;
- le langage des mots ayant moins de a que de b ;
- les mots de Dyck, mots bien parenthésés sur un alphabet de parenthèses : `([()[]])()`.

`(   [   (   )   [   ]   ]   )   (   )`

Exemples

- le langage des mots ayant autant de a que de b ;
- le langage des palindromes ;
- le langage des mots ayant moins de a que de b ;
- les mots de Dyck, mots bien parenthésés sur un alphabet de parenthèses : `([()[]])()`.



Exemple de preuve plus compliquée

Exemple

Montrer que $L = \{a^{n^2}\}$ n'est pas régulier.

Observation 1

$L = \{\varepsilon, a, a^4, a^9, a^{16}, \ldots\}$.
Notons $w_n = a^{n^2}$ le $n^{\text{ème}}$ mot de cette suite.
Calculons $|w_{n+1}| - |w_n| = (n+1)^2 - n^2 = 2n + 1$. La différence entre deux mots consécutifs augmente de façon linéaire avec n.

Exemple de preuve plus compliquée

Observation 2

Soit $w \in \Sigma^*$, et soit une factorisation de w en trois sous-mots : $w = xyz$.

Considérons la suite $\{xz, xyz, xy^2z, xy^3z, xy^4z, ...\}$. La différence du nombre de caractères entre deux mots consécutifs de cette suite est constante puisqu'elle est toujours égale à $|y|$.

Preuve

Soit $\mathcal{A} = (\Sigma, Q, q_0, F, T)$ l'automate reconnaissant L. On choisit un mot w_n tel que :

- $|w_n| \geq |Q|$
- $w_n = xyz$ avec $|y| \leq |Q|$
- $2n + 1 > |Q|$

Exemple de preuve plus compliquée

Fin de la preuve

En prenant $N = |Q|$, quelque soit la factoristaion $w_n = xyz$. Avec l'observation 1, on a que si xy^2z appartient au langage alors :

$$|xy^2z| - |xyz| \geq 2n + 1 > |Q|$$

Et par l'observation 2 on a :

$$|xy^2z| - |xyz| = |y| \leq |Q|$$

Définition

Une $\boxed{\textbf{grammaire}}$ $\mathcal{G} = (\Sigma, V, S, \mathcal{P})$ telle que :

- Σ un alphabet fini de symboles $\boxed{\textbf{terminaux}}$;
- V un alphabet fini de symboles $\boxed{\textbf{non terminaux}}$ ou $\boxed{\textbf{variables}}$, $V \cap \Sigma = \emptyset$;
- $S \in V$ appelé $\boxed{\textbf{axiome}}$;
- $\mathcal{P} \subseteq (V \cup \Sigma)^* \times (V \cup \Sigma)^*$ l'ensemble des $\boxed{\textbf{rêgles de production}}$.

Exemple

$\Sigma = \{a, b\}$, $V = \{S\}$ et
$\mathcal{P} = \{S \to aSb, S \to \varepsilon\}$

Cette grammaire génère le langage $a^n b^n$.

Grammaire

Exemple

$\Sigma = \{a\},\ V = \{S\}$ et $\mathcal{P} = \{$

$S \to aaS,$

$S \to \varepsilon\ \}$

Langage des mots avec un nombre pair de a.

Exemple

$\Sigma = \{a, b, c\},\ V = \{S, X\}$ et les règles de $\mathcal{P} = \{$

$S \to aS,$

$S \to bS,$

$S \to cS,$

$S \to aX,$

$X \to b\ \}$

Langage des mots qui se terminent par ab.

Écriture plus compacte : $\mathcal{P} = \{S \to (a + b + c)S \quad | \quad ab\ \}$.

Hiérarchie de Chomski

Soit une grammaire $\mathcal{G} = (\Sigma, V, S, \mathcal{P})$:

Type	Langage	Grammaire	Machine
3	rationnel, régulier ou reconnaissable	régulière droite $X \to a, X \to aY, X \to \varepsilon$ $X, Y \in V,\ a \in \Sigma$ (régulière gauche)	automate fini
2	algébrique ou hors-contexte	algébrique ou hors-contexte $X \to \alpha$ $X \in V,\ \alpha \in (\Sigma \cup V)^*$	automate à pile
1	contextuel	contextuelle $\alpha X \beta \to \alpha \gamma \beta$ $X \in V,\ \alpha, \beta, \gamma \in (\Sigma \cup V)^*$	machine de Turing bornée
0	récursivement énumérable	générales $\alpha \to \beta$ $\alpha, \beta \in (\Sigma \cup V)^*$	machine de Turing