

Chapitre IV : Langages rationnels

Goulven GUILLOU

UBO – Département Informatique



1 / 24

Quels sont les langages reconnaissables ?

On sait que l'ensemble des langages reconnaissables est fermé par union (+), concaténation (.) et fermeture itérative (*) et contient les langages finis.

On note $Rec(A)$ ou $Rec(A^*)$ l'ensemble des langages reconnaissables sur l'alphabet A .

Donc les langages construits à partir d'un alphabet et des opérateurs d'union, de concaténation et de fermeture itérative sont reconnaissables.

Y en a-t-il d'autres ?

2 / 24

Les langages rationnels

Soit A un alphabet, la classe des langages *rationnels* ou *réguliers* sur l'alphabet A , notée $Rat(A^*)$ ou Rat est le plus petit ensemble satisfaisant les conditions suivantes :

- $\emptyset \in Rat$.
- $x \in Rat, \forall x \in A$.
- Soient $L \in Rat$ et $L' \in Rat$ alors $L + L' \in Rat, L.L' \in Rat$ et $L^* \in Rat$.

3 / 24

Expressions régulières

L'expression d'un langage selon une décomposition en unions, produits et étoiles s'appelle une expression régulière du langage. Une expression régulière est construite à partir de caractères, des opérateurs binaires de concaténation et d'union, de l'opérateur unaire * et de parenthèses.

Par exemple : $(a+b)^*abb(a+b)^*$

Un langage est *rationnel* ou *régulier* ssi il est dénoté par une expression régulière.

Voir le chapitre 3 pour les priorités permettant de supprimer des parenthèses.

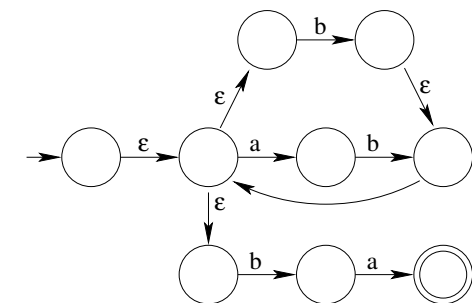
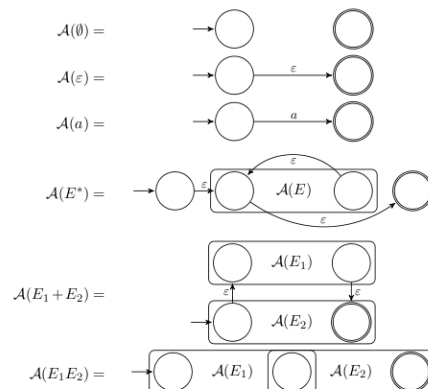
4 / 24

- sur un alphabet A , le langage L constitué des mots de longueur paire est régulier car il est décrit par l'expression $L = (AA)^*$.
- sur $A = \{0, 1\}$, le langage L des mots commençant par 1 est décrit par l'expression régulière $L = 1(1 + 0)^*$. De même $L' = 0^*10^*10^*$ est régulier et constitué des mots contenant exactement trois fois la lettre 1.
- sur $A = \{a, b, c, \dots, z\}$, le langage des mots contenant comme facteur *facteur* ou *factrice* est dénoté par $A^*fact(eur + rice)A^*$ et est donc régulier.
- sur $A = \{a, b\}$, $L = \{a^n b^n | n \in \mathbb{N}\}$ n'est pas régulier.

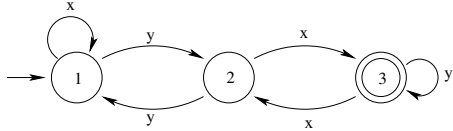
Un langage est régulier si et seulement si il est accepté par un automate fini. Autrement dit :

$$Rat(A^*) = Rec(A^*)$$

Les automates générés ont un unique état initial sans transition entrante et un unique état accepteur sans transition sortante. On procède par induction sur l'expression.



$W_{p,E,q}$ = ensemble des mots permettant de passer de p à q en n'utilisant que les états intermédiaires de $E \subseteq Q$.



$$\begin{aligned} W_{1,\{1,2,3\},3} &= W_{1,\{2,3\},3} \cup W_{1,\{2,3\},1} \cdot W_{1,\{2,3\},1}^* \cdot W_{1,\{2,3\},3} \\ &= W_{1,\{2,3\},1}^* \cdot W_{1,\{2,3\},3} \\ &= W_{1,\{2,3\},1}^* \cdot W_{1,\{2\},3} \cdot W_{3,\{2\},3}^* \end{aligned}$$

$$W_{1,\{2\},3} = \{yx\}$$

$$W_{3,\{2\},3} = W_{3,\emptyset,3} \cup W_{3,\emptyset,2} W_{2,\emptyset,3}^* W_{2,\emptyset,3} = \{\varepsilon, y, xx\}$$

$$W_{1,\{2,3\},1} = W_{1,\{2\},1} \cup W_{1,\{2\},3} \cdot W_{3,\{2\},3}^* \cdot W_{3,\{2\},1}$$

$$W_{1,\{2\},1} = \{\varepsilon, x, yy\}$$

$$W_{3,\{2\},1} = \{xy\}$$

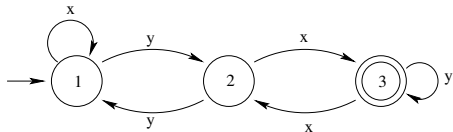
$$\text{d'où } L = (x + yy + yx(y + xx)^* xy)^* yx(y + xx)^*$$

Proposition 1 : Soient U et V deux langages rationnels tels que $\varepsilon \notin U$ alors l'unique langage solution de $L = LU + V$ est VU^* .

On a la proposition symétrique :

Proposition 2 : Soient U et V deux langages rationnels tels que $\varepsilon \notin U$ alors l'unique langage solution de $L = UL + V$ est U^*V .

On note L_i le langage de l'automate ayant le même état initial que l'automate de départ avec i pour unique état accepteur.



$$\begin{cases} L_1 = L_1x + L_2y + \varepsilon \text{ (car 1 état initial)} \\ L_2 = L_1y + L_3x \\ L_3 = L_2x + L_3y \text{ (langage que l'on cherche)} \end{cases}$$

Attention dans le cas où l'automate a plusieurs états accepteurs, faire l'union des L_i correspondants !

Elimination à partir de L_3 jusqu'à L_1 .

$$\begin{cases} L_1 = L_1x + L_2y + \varepsilon \\ L_2 = L_1y + L_3x \\ L_3 = L_2x + L_3y \end{cases} \quad \begin{cases} L_1 = L_1x + L_2y + \varepsilon \\ L_2 = L_1y + L_2xy^*x \\ L_3 = L_2xy^* \end{cases}$$

$$\begin{cases} L_1 = L_1x + L_1y(xy^*x)^*y + \varepsilon \\ L_2 = L_1y(xy^*x)^* \\ L_3 = L_2xy^* \end{cases} \quad \begin{cases} L_1 = (x + y(xy^*x)^*y)^* \\ L_2 = L_1y(xy^*x)^* \\ L_3 = L_2xy^* \end{cases}$$

D'où $L_2 = (x + y(xy^*x)^*y)^*y(xy^*x)^*$
et

$$L_3 = (x + y(xy^*x)^*y)^*y(xy^*x)^*xy^*$$

Elimination de L_1 vers L_3 (ce que l'on cherche).

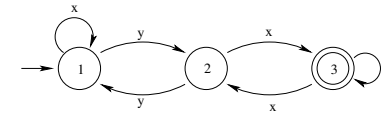
$$\begin{cases} L_1 = L_1x + L_2y + \varepsilon \\ L_2 = L_1y + L_3x \\ L_3 = L_2x + L_3y \end{cases} \quad \begin{cases} L_1 = (L_2y + \varepsilon)x^* \\ L_2 = (L_2y + \varepsilon)x^*y + L_3x \\ L_3 = L_2xy^* \end{cases}$$

$$\begin{cases} L_1 = (L_2y + \varepsilon)x^* \\ L_2 = (x^*y + L_3x)(yx^*y)^* \\ L_3 = (x^*y + L_3x)(yx^*y)^*xy^* \end{cases}$$

D'où

$$L_3 = x^*y(yx^*y)^*xy^*(x(yx^*y)^*xy^*)^*$$

On note L_i le langage de l'automate de départ ayant i pour état initial.



$$\begin{cases} L_1 = xL_1 + yL_2 \text{ (langage que l'on cherche)} \\ L_2 = yL_1 + xL_3 \\ L_3 = xL_2 + yL_3 + \varepsilon \text{ (car 3 état accepteur)} \end{cases}$$

On résout par élimination de L_3 jusqu'à L_1 :

$$\begin{cases} L_1 = xL_1 + yL_2 \\ L_2 = yL_1 + xy^*xL_2 + xy^* \\ L_3 = y^*(xL_2 + \varepsilon) \end{cases} \quad \begin{cases} L_1 = (x + y(xy^*x)^*y)L_1 \\ \quad + y(xy^*x)^*xy^* \\ L_2 = (xy^*x)^*(yL_1 + xy^*) \\ L_3 = y^*(xL_2 + \varepsilon) \end{cases}$$

D'où $L_1 = (x + y(xy^*x)^*y)^*y(xy^*x)^*xy^*$

Le même résultat que la première résolution de la méthode 1 !

$Rat(X^*)$ est fermé par union, intersection, complémentaire et fermeture itérative.

Sous UNIX on utilise des expressions régulières dans au moins 3 types de commandes :

- les éditeurs UNIX comme `vi` ou `emacs`.
- le programme d'équivalence de motifs `grep` et ses cousins.
- l'analyse lexicale avec la commande UNIX `flex`.

[aotu] désigne l'ensemble des lettres apparaissant dans auto.
 [aotu]* désigne l'ensemble des mots composés des lettres a, o, t, u
 uniquement.
 [A-Za-z] indique l'ensemble des lettres de l'alphabet en minuscules
 ou majuscules.
ATTENTION [-+*/] est différent de [+*/*] !
 [^aotu] désigne n'importe quel caractère excepté a, o, t ou u.

Le symbole ^ indique le début d'une ligne, et \$ indique la fin d'une
 ligne.

Utilisation : `grep '^[aotu]*\$' monfichier` affichera tous les
 mots de monfichier qui tiennent sur une ligne et composés
 uniquement des lettres appartenant au mot auto.

Le caractère . remplace un caractère quelconque hormis le caractère
 de passage à la ligne.

L'expression régulière : `. *a.*e.*i.*o.*u.`
 décrit toutes les chaînes contenant les voyelles dans l'ordre.

Avec `grep` il suffit de faire : `grep a.*e.*i.*o.*u`

`R?` signifie "au plus une fois R".

`R+` équivaut à `RR*`

`[-]?[0-9]+` désigne un entier signé.

On peut faire référence à toute partie d'une expression régulière
 encadrée par `\(` et `\)` par `\n` n étant l'ordre d'apparition.

`grep '^\(.\).*\1$' monfichier.txt` va rechercher toutes les
 lignes qui commencent et finissent par le même caractère.

Deux ensembles ont même *cardinal* ou *cardinalité* s'il existe une bijection entre les deux.

"Avoir la même cardinalité" est une relation d'équivalence.

On appelle n la cardinalité de $\{0, 1, \dots, n-1\}$.

Définition : Un ensemble est dénombrable s'il est fini ou s'il existe une bijection entre lui et \mathbb{N} .

Théorème : L'ensemble des sous-ensembles d'un ensemble infini dénombrable n'est pas dénombrable.

preuve :

	a_0	a_1	a_2	a_3	a_4	...
s_0	×	×		×		
s_1	×	Δ		×		
s_2		×	×		×	
s_3	×		×	Δ		
s_4		×		×	Δ	

Les expressions régulières sont dénombrables.

Les langages réguliers sont donc dénombrables.

Les langages ne sont pas dénombrables car c'est l'ensemble des parties d'un ensemble dénombrable.

Conclusion : il y a plus de langages que de langages réguliers.

Soit L un langage régulier infini. Alors il existe un entier $m > 0$ tel que tout mot $w \in L$ de longueur supérieure ou égale à m se décompose en 3 parties : $w = xyz$ avec $|y| \geq 1$ et $xy^kz \in L$ pour tout $k \in \mathbb{N}$.

Exemple : $\{a^n b^n, n \in \mathbb{N}\}$ n'est pas régulier.

Supposons $\{a^n b^n, n \in \mathbb{N}\}$ régulier. Alors il existe un entier $m > 0$ tel que tout mot de longueur au moins m soit décomposable. En particulier $w = a^m b^m$ est décomposable.

On a trois cas possibles :

- $y = a^s, s > 0$
- $y = a^s b^t, s > 0$ et $t > 0$
- $y = b^t, t > 0$

Dans aucun de ces cas xy^2z n'appartient à $\{a^n b^n, n \in \mathbb{N}\}$.