6. Langages rationnels : passages d'une représentation à une autre

Rappel :

Théorème

Un langage L est rationnel (classe 3)
ssi
il existe une grammaire régulière G telle que $L(G) = L$
ssi
il existe un automate d'états finis reconnaissant L
ssi
il existe une expression régulière engendrant L

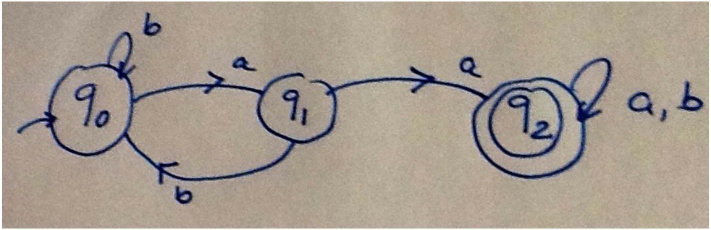
6.1. Passages d'une représentation à une autre

La preuve est constructive, sous forme d'algorithmes de passage :

Automate d'états finis$\leftrightarrow$Grammaire régulière
$\updownarrow$
Expression régulière

Automate d'états finis → Grammaire régulière :

Exemple :

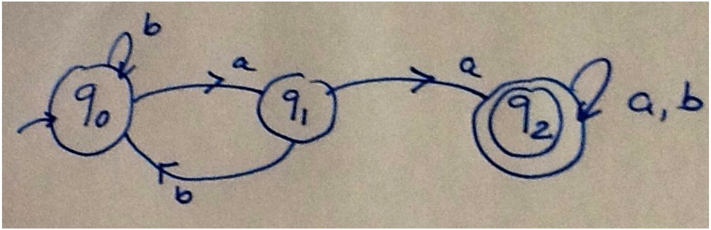


6.2. Automate d'états finis $\leftrightarrow$ Grammaire régulière

On représente l'état initial q_0 par l'axiome S
On représente chaque autre état Q_i par un non-terminal A_i
Si Q_i est un état final, on ajoute une ϵ -transition $A_i \rightarrow \epsilon$

Automate d'états finis → Grammaire régulière :

Exemple :



$S \rightarrow bS|aA_1$

$A_1 \rightarrow bS|aA_2$

$A_2 \rightarrow aA_2|bA_2|\epsilon$

Cas particuliers :

$S \rightarrow bS|aA$

$A \rightarrow bB$

$B \rightarrow aB|\epsilon$

peut se réécrire :

$S \rightarrow bS|abB$

$B \rightarrow aB|\epsilon$

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

peut se réécrire :
$S \rightarrow bS|aA$
$A \rightarrow aA|b$

Grammaire régulière $\rightarrow$ Automate d'états finis :

On réécrit la grammaire régulière pour obtenir des règles de production de la forme :

$X \rightarrow aY$ ou $X \rightarrow \epsilon$, avec $X, Y \in N$ et $a \in T$

6.3. Grammaire régulière $\rightarrow$ Automate d'états finis

Exemple 1 :
$S \rightarrow aaA$
$A \rightarrow bA|\epsilon$

se réécrit :
$S \rightarrow aA_1$
$A_1 \rightarrow aA_2$
$A_2 \rightarrow bA_2|\epsilon$

6.3. Grammaire régulière $\rightarrow$ Automate d'états finis

Exemple 2 :

$S \rightarrow aS|a$

se réécrit :
$S \rightarrow aS|aA_1$
$A_1 \rightarrow \epsilon$

6.3. Grammaire régulière $\rightarrow$ Automate d'états finis

On représente ensuite l'axiome S par l'état q_0
et le non-terminal A_i par l'état q_i
Les états finaux correspondent aux règles de la forme ;
$A_i \rightarrow \epsilon$.

Exemple :

$S \rightarrow aaA$

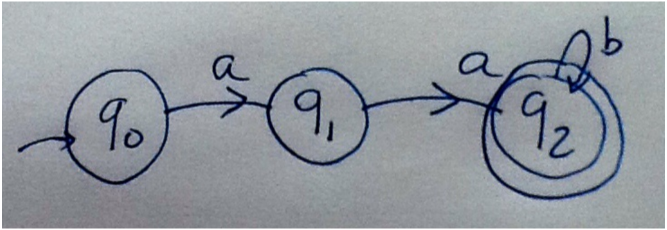
$A \rightarrow bA|\epsilon$

se réécrit :

$S \rightarrow aA_1$

$A_1 \rightarrow aA_2$

$A_2 \rightarrow bA_2|\epsilon$



Rappels :
Définition inductive des expressions régulières :

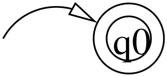
Définition

Définition inductive des expressions régulières :

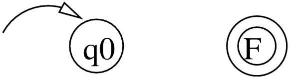
- *Base :* $\emptyset$, ϵ *et les caractères de* Σ *sont des expressions régulières, représentant respectivement les langages* $\emptyset$,$\{\epsilon\}$, $\{x\}$ *si* $x \in \Sigma$.
- *Règles : si r et s sont des expressions régulières représentant les langages R et S, alors* $(r + s)$, $r.s$, r^*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

Pour construire un AEF, on applique cette définition inductive :
base :

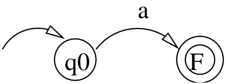
r=ϵ



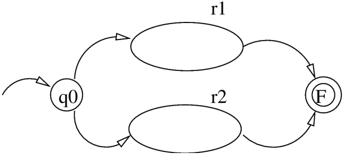
$\emptyset$



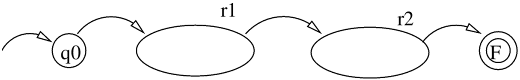
'a'



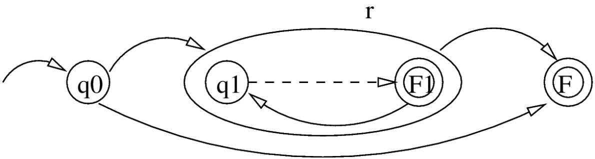
règles de construction de l'automate :



$r_1 + r_2$



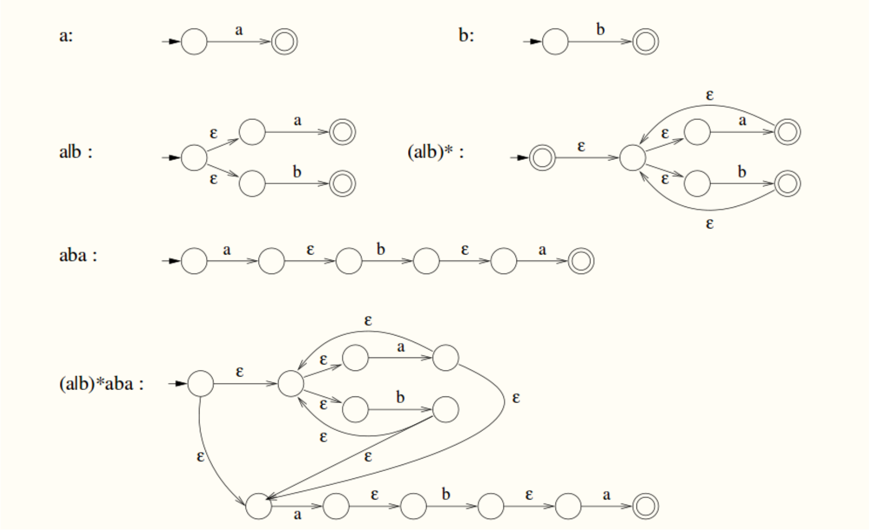
$r_1 r_2$



r^*

Pour construire r^+, on fera rr^*

Exemple :
$(a+b)^* aba$



6.5. Automate d'états finis$\rightarrow$ Expression régulière

Algorithme de Mac Naughton et Yamada :

1. Transformation de l'automate en automate généralisé
2. Transformation de l'automate généralisé en expression régulière

Bibliographie et figures : Cours d'Alexis Nasr

6.5. Automate d'états finis $\rightarrow$ Expression régulière

Définition

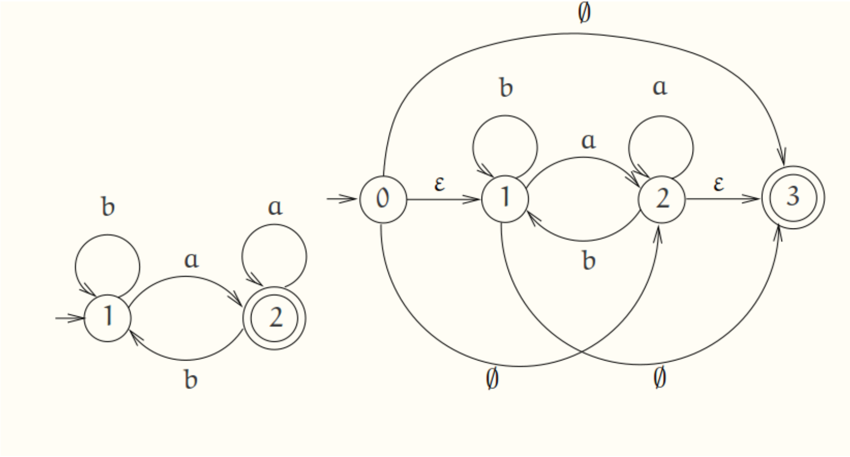
Automate généralisé : les transitions sont étiquetées par des expressions régulières (ou par $\emptyset$).

6.5. Automate d'états finis$\rightarrow$ Expression régulière

Transformation d'un automate en automate généralisé :
1. ajouter un nouvel état initial avec une ϵ- transition vers q_0
2. ajouter un nouvel état final vers lequel les anciens états finaux sont envoyés par une ϵ- transition
3. des transitions étiquetées par $\emptyset$ sont ajoutées entre les états qui ne sont reliés par aucune transition, mais entre lesquels il existe un chemin dans l'automate de départ
(ces transitions ne peuvent pas être franchies)

6.5. Automate d'états finis→ Expression régulière

Exemple d'initialisation à un automate généralisé :

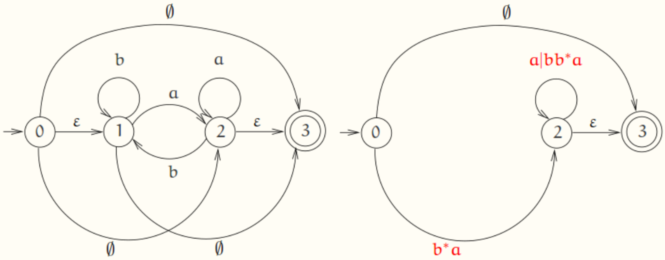


6.5. Automate d'états finis$\rightarrow$ Expression régulière

A chaque itération, on supprime un état.
A la fin, il reste une seule transition de l'état initial à l'état final, étiquetée par l'expression régulière recherchée.

6.5. Automate d'états finis→ Expression régulière

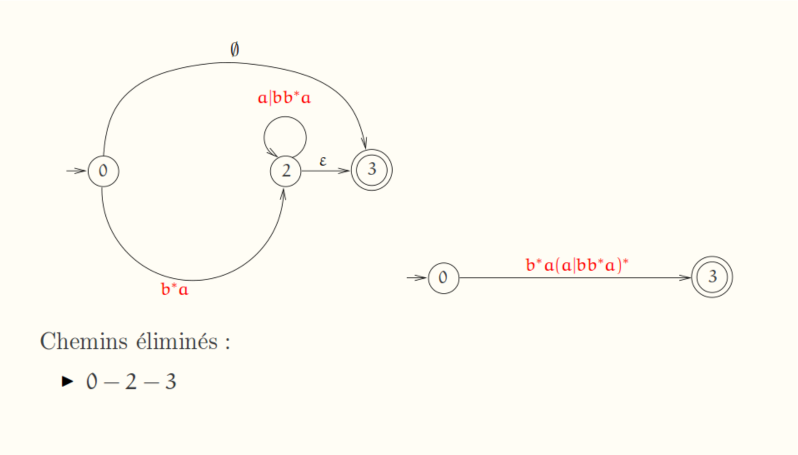
Exemple : on supprime l'état 1 :



Chemins éliminés :

- 0 − 1 − 2
- 2 − 1 − 2
- 0 − 1 − 3

Exemple (suite) : on supprime l'état 2 :



On obtient l'expression régulière $b^*a(a|bb^*a)^*$

6.6. Applications

- Montrer qu'un langage est rationnel
- Montrer que deux expressions régulières sont équivalentes
- Trouver une grammaire régulière
- Améliorer une grammaire régulière
- ...

6.7. Hiérarchie de Chomsky

	Nom	Type de production	Machines acceptant ce lge
0	Lges récursivement énumérables	X→Y $X \in \mathbb{N}^+$, Y∈(N∪T)*	Machines de Turing Automates à plusieurs piles
1	Lges contextuels	X→Y $X \in \mathbb{N}^+$, Y∈(N∪T)* , |Y|≥|X|	Machines de Turing bornées
2	Lges context-free ou algébriques	X→Y $X \in \mathbb{N}$, Y∈(N∪T)*	Automates à pile
3	Lges rationnels (réguliers)	X→Y $X \in \mathbb{N}$, Y=tA ou Y=t	Automates d'états finis

6.7. Hiérarchie de Chomsky

Dans la 2e partie de ce cours, vous étudierez :

- existence de langages non rationnels
- les langages algébriques et les automates à pile
- existence de langages non algébriques ...