Théorie des langages

Alexis Nasr

Théorème de Kleene

- $L \subset \Sigma^*$ est un langage régulier si et seulement si il est défini par un automate fini.
- Preuve par construction :
  - méthode pour constuire un automate fini à partir d'une expression régulière (Algorithme de Thompson)
  - deux méthodes permettant de construire une expression régulière à partir d'un automate fini.
    - Automates généralisés
    - Système d'équations régulières

Expressions régulières

Elles permettent de dénoter des langages réguliers sur Σ.

1. $\emptyset$ est une expression régulière dénotant le langage régulier $\emptyset$.
2. ε est une expression régulière dénotant le langage régulier $\{\varepsilon\}$.
3. a (tel que $a \in \Sigma$) est une expression régulière dénotant le langage régulier $\{a\}$.

Si p et q sont des expressions régulières dénotant respectivement les ensembles réguliers P et Q alors :

5. $(p + q)$ est une expression régulière dénotant le langage régulier $P \cup Q$
6. (pq) est une expression régulière dénotant le langage régulier PQ
7. $(p)^*$ est une expression régulière dénotant le langage régulier P^*

$E = \emptyset$

1. $E = \emptyset$. E dénote le langage $\emptyset$ qui est aussi le langage reconnu par l'automate suivant : $A = \langle \{q\}, \Sigma, \delta, q, \emptyset \rangle$.

$E = \varepsilon$

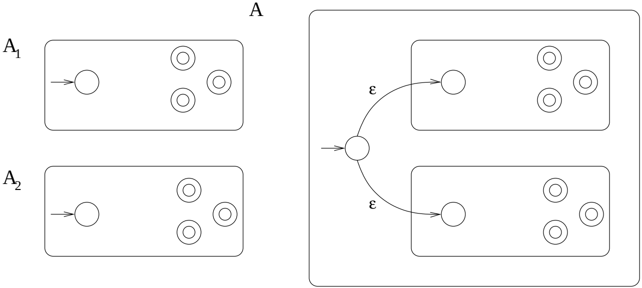
1. $E = \varepsilon$. E dénote le langage $\{\varepsilon\}$ qui est aussi le langage reconnu par l'automate suivant :$A = \langle \{q\}, \Sigma, \delta, q, \{q\} \rangle$ avec $\delta(q, x) = \emptyset, \forall x \in \Sigma$.

$E = a$

1. $E = a$ avec $a \in \Sigma$. E dénote le langage $\{a\}$ qui est aussi le langage reconnu par l'automate suivant : $A = \langle \{q_1, q_2\}, \Sigma, \delta, q_1, \{q_2\} \rangle$ avec $\delta(q_1, a) = q_2$, $\delta(r, x) = \emptyset$ pour $r \in Q - \{q_1\}$ ou $x \in \Sigma - \{a\}$.

$E = E_1 + E_2$

4. $E = E_1 + E_2$
Si $L(E_1) = L(A_1)$ et $L(E_2) = L(A_2)$ alors $L(E_1 + E_2) = L(A)$ où A est défini de la façon suivante :



$E = E_1 + E_2$

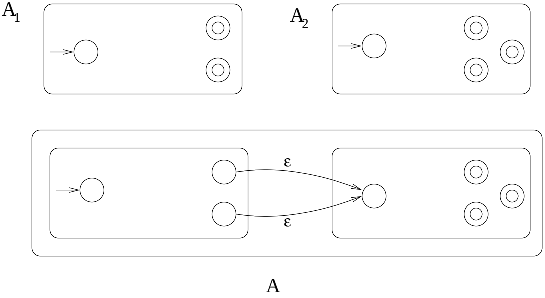
$A = \langle Q, \Sigma, \delta, q_0, F \rangle$
$A_1 = \langle Q_1, \Sigma_1, \delta_1, q_1, F_1 \rangle$
$A_2 = \langle Q_2, \Sigma_2, \delta_2, q_2, F_2 \rangle$

1. $Q = \{q_0\} \cup Q_1 \cup Q_2$. L'ensemble des états de A est constitué des états de A_1, des états de A_2 et d'un nouvel état initial q_0.
2. l'état q_0 est l'état initial de A.
3. $F = F_1 \cup F_2$. L'ensemble des états d'acceptation A est constitué des états d'acceptation de A_1 et des états d'acceptation de A_2.
4. La fonction de transition δ est définie de la façon suivante pour tout $q \in Q$ et tout $a \in \Sigma_\varepsilon$:

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & \text{si } q \in Q_1 \\ \delta_2(q,a) & \text{si } q \in Q_2 \\ \{q_1, q_2\} & \text{si } q = q_0 \text{ et } a = \varepsilon \\ \emptyset & \text{si } q = q_0 \text{ et } a \neq \varepsilon \end{cases}$$

$E = E_1 E_2$

5. $E = E_1 E_2$
Si $L(E_1) = L(A_1)$ et $L(E_2) = L(A_2)$ alors $L(E_1 E_2) = L(A)$ où A est défini de la façon suivante :



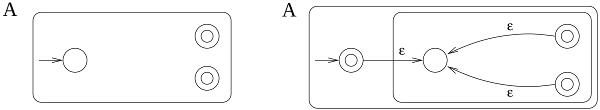
$E = E_1E_2$

1. $Q = Q_1 \cup Q_2$. L'ensemble des états de A est constitué des états de A_1 et des états de A_2.
2. $q_0 = q_1$. L'état initial q_0 est le même que celui de A_1.
3. $F = F_2$. L'ensemble des états d'acceptation celui de A_2.
4. La fonction de transition δ est définie de la façon suivante pour tout $q \in Q$ et tout $a \in \Sigma_\varepsilon$:

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & \text{si } q \in Q_1 \text{ et } q \notin F_1 \\ \delta_1(q,a) & \text{si } q \in F_1 \text{ et } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\} & \text{si } q \in F_1 \text{ et } a = \varepsilon \\ \delta_2(q,a) & \text{si } q \in Q_2 \end{cases}$$

$E = E_1^*$

6. $E = E_1^*$
Si $L(E_1) = L(A_1)$ $L(E_1^*) = L(A)$ où A est défini de la façon suivante :



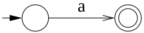
$E = E_1^*$

1. $Q = \{q_0\} \cup Q_1$. L'ensemble des états de A_1 plus un nouvel état initial q_0.
2. $q_0 = q_1$. L'état q_0 est l'état initial de A_1^*.
3. $F = \{q_0\} \cup F_1$. L'ensemble des états d'acceptation A est constitué des états d'acceptation de A_1 et du nouvel état initial.
4. La fonction de transition δ est définie de la façon suivante pour tout $q \in Q$ et tout $a \in \Sigma_\varepsilon$:

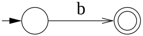
$$\delta(q,a) = \begin{cases} \delta_1(q,a) & \text{si } q \in Q_1 \text{ et } q \notin F_1 \\ \delta_1(q,a) & \text{si } q \in F_1 \text{ et } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_1\} & \text{si } q \in F_1 \text{ et } a = \varepsilon \\ \{q_1\} & \text{si } q = q_0 \text{ et } a = \varepsilon \\ \emptyset & \text{si } q = q_0 \text{ et } a \neq \varepsilon \end{cases}$$

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

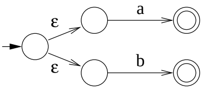
a:



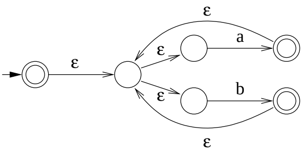
b:



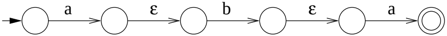
a|b :



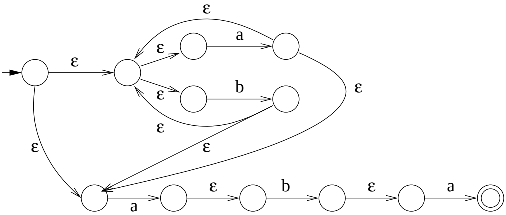
(a|b)* :



aba :



(a|b)*aba :



Automate $\Rightarrow$ expression régulière

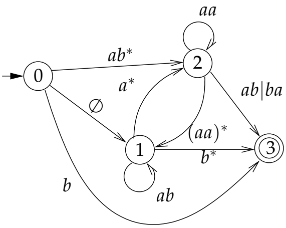
Méthode des automates généralisés
Deux étapes :

- Transformation de l'automate en automate généralisé.
- Transformation de l'automate généralisé en expression régulière.

Automate généralisé

- Un automate généralisé est un automate fini dont les transitions sont étiquetées par des expressions régulières et non pas simplement des symboles ou le mot vide.
- L'automate généralisé lit le mot à reconnaître par blocs de symboles.
- On impose trois contraintes supplémentaires dans notre cas :
  1. L'état initial possède des transitions vers tous les autres états, mais aucun état n'a de transition vers l'état initial.
  2. Il n'y a qu'un état d'acceptation qui ne possède aucune transition vers d'autres états, mais tous les autres états possèdent une transition vers l'état d'acceptation. De plus, l'état d'acceptation est distinct de l'état initial.
  3. A l'exception de l'état d'acceptation et de l'état initial, tous les états possèdent une transition et une seule vers tous les autres états.

Exemple d'automate généralisé



Transformation d'un automate en automate généralisé

1. ajouter un nouvel état initial possèdant une transition-ε vers l'ancien état initial.

Transformation d'un automate en automate généralisé

1. ajouter un nouvel état initial possèdant une transition-ε vers l'ancien état initial.
2. ajouter un nouvel état d'acceptation vers lequel il existe une transition-ε partant des anciens états d'acceptation.

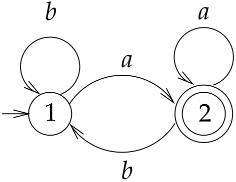
Transformation d'un automate en automate généralisé

1. ajouter un nouvel état initial possèdant une transition-ε vers l'ancien état initial.
2. ajouter un nouvel état d'acceptation vers lequel il existe une transition-ε partant des anciens états d'acceptation.
3. S'il existe plusieurs transitions entre deux états, elles sont remplacées par une transition unique étiquetée par l'union des étiquettes des différentes transitions.

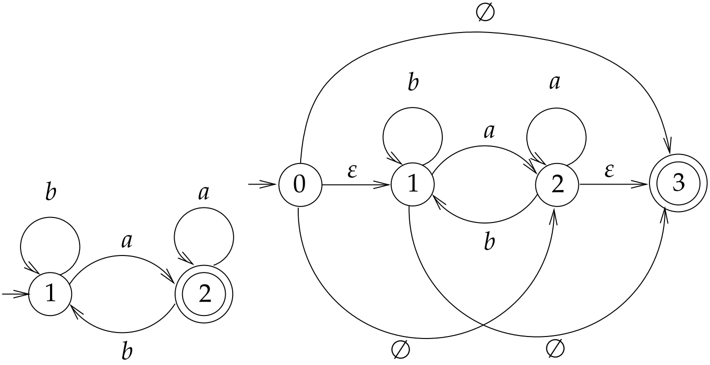
Transformation d'un automate en automate généralisé

1. ajouter un nouvel état initial possèdant une transition-ε vers l'ancien état initial.
2. ajouter un nouvel état d'acceptation vers lequel il existe une transition-ε partant des anciens états d'acceptation.
3. S'il existe plusieurs transitions entre deux états, elles sont remplacées par une transition unique étiquetée par l'union des étiquettes des différentes transitions.
4. Des transitions étiquetées $\emptyset$ sont ajoutées entre les états qui ne sont reliés par aucune transition. Cet ajout ne modifie pas le langage reconnu par l'automate car une transition étiquetée $\emptyset$ ne peut jamais être franchie.

Transformation d'un automate en automate généralisé



Transformation d'un automate en automate généralisé

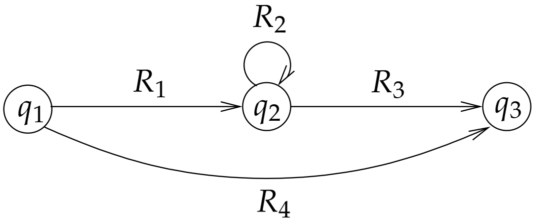


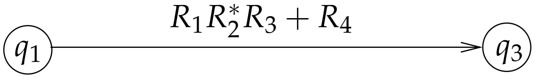
Réduction de l'automate généralisé

Procédure itérative

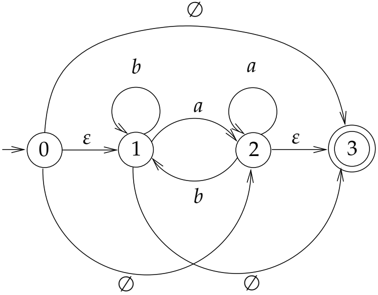
- à chaque itération, on diminue d'un le nombre d'états de l'automate généralisé.
- A l'issue du processus, on obtient un automate généralisé comportant deux états (l'état initial et l'état d'acceptation) et une transition entre les deux.
- L'expression régulière étiquetant cette transition dénote le langage de l'automate initial.

Principe de réduction





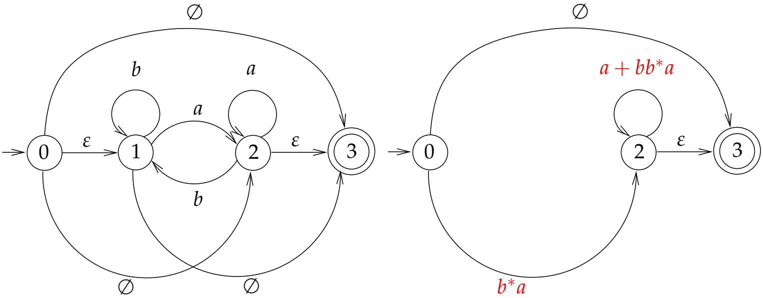
Exemple



Elimination de l'état 1
Chemins éliminés :

- $0 - 1 - 2$
- $2 - 1 - 2$
- $0 - 1 - 3$

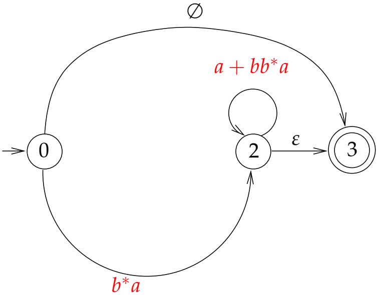
Exemple



Elimination de l'état 1
Chemins éliminés :

- $0 - 1 - 2$
- $2 - 1 - 2$
- $0 - 1 - 3$

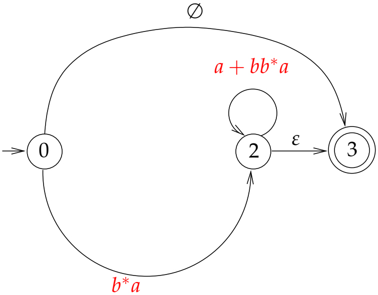
Exemple

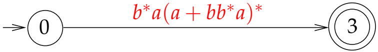


Elimination de l'état 2
Chemins éliminés :

- $0 - 2 - 3$

Exemple





Elimination de l'état 2
Chemins éliminés :

- $0 - 2 - 3$

Automate ⇒ expression régulière

Méthode du système d'équations régulières

- On écrit un système d'équations correspondant à l'automate A.
- La résolution du système d'équations permet de trouver l'expression régulière dénotant le langage reconnu par A.

Langages L_q

- Pour chaque état q d'un automate $A = \langle Q, \Sigma, \delta, q_0, F \rangle$, on note L_q l'ensemble des mots w tels que le calcul de A sur w à partir de q aboutit à un état d'acceptation.
- Autrement dit :

$$L_q = \{w \in \Sigma^* \mid (q, w) \overset{*}{\vdash} (q_f, \varepsilon) \text{ avec } q_f \in F\}$$

- En particulier : $L_{q_0} = L(A)$

Système d'équations associé à un automate

- Supposons que $\Sigma = \{a, b\}$
- Supposons que $\delta(p, a) = q$ et $\delta(p, b) = r$
- Si $p \notin F$ alors $m \in L_p$ si
  - soit $m = au$ avec $u \in L_q$
  - soit $m = bu$ avec $u \in L_r$
- par conséquent, on a :

$$L_p = aL_q \cup bL_r$$

- si $p \in F$ alors :

$$L_p = aL_q \cup bL_r \cup \{\varepsilon\}$$

Système d'équations associé à un automate

- On note X_p une expression régulière qui dénote L_p, on obtient :

$$\begin{array}{lcll} X_p & = & aX_q + bX_r & \text{si } p \notin F \\ X_p & = & aX_q + bX_r + \varepsilon & \text{si } p \in F \end{array}$$

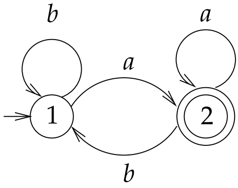
- Généralisation :
  - étant donné un automate déterministe $A = \langle Q, \Sigma, \delta, q_0, F \rangle$, et $p \in Q$.
  - Si les transitions ayant p pour origine sont : $\delta(p, a_1) = p_1 \ldots \delta(p, a_k) = p_k$, alors les expressions régulières qui dénotent les langages $L_p, L_{p_1} \ldots, L_{p_k}$ sont liées par l'équation :

    $$X_p = a_1 X_{p_1} + \ldots + a_k X_{p_k} \text{ si } p \notin F$$

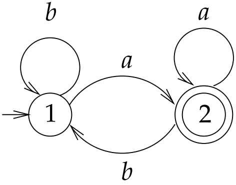
    ou par l'équation

    $$X_p = a_1 X_{p_1} + \ldots + a_k X_{p_k} + \varepsilon \text{ si } p \in F$$

Exemple



Exemple



$$\begin{aligned} X_1 &= bX_1 + aX_2 \\ X_2 &= aX_2 + bX_1 + \varepsilon \end{aligned}$$

Lemme d'Arden

Soient A, B deux langages sur Σ tels que $\varepsilon \notin A$. Alors l'équation

$$X = AX \cup B$$

admet A^*B comme unique solution.

Preuve

1. A^*B est une solution de l'équation $X = AX \cup B$.
2. S'il existe une autre solution X alors est équivalente à A^*B $(L(X) = L(A^*B))$

Preuve

1. A^*B est une solution de l'équation $X = AX \cup B$.
   En effet : $AA^*B \cup B = (AA^* \cup \{\varepsilon\})B = A^*B$
2. S'il existe une autre solution X alors est équivalente à A^*B $(L(X) = L(A^*B))$

Preuve

1. A^*B est une solution de l'équation $X = AX \cup B$.
   En effet : $AA^*B \cup B = (AA^* \cup \{\varepsilon\})B = A^*B$
2. S'il existe une autre solution X alors est équivalente à A^*B $(L(X) = L(A^*B))$
   1. $L(A^*B) \subseteq L(X)$
      En effet : si X est une solution alors

$$
\begin{array}{rcl}
X & = & AX \cup B \\
X & = & A(AX \cup B) \cup B \\
X & = & A^2X \cup AB \cup B \\
 & \cdots & \\
X & = & A^{n+1}X \cup A^nB \cup A^{n-1}B \cup \ldots \cup AB \cup B \ \forall n \geq 0
\end{array}
$$

      X contient donc tous les A^nB, $\forall n \geq 0$ et donc $L(A^*B) \subseteq L(X)$

Preuve

1. A^*B est une solution de l'équation $X = AX \cup B$.
   En effet : $AA^*B \cup B = (AA^* \cup \{\varepsilon\})B = A^*B$
2. S'il existe une autre solution X alors est équivalente à A^*B $(L(X) = L(A^*B))$
   1. $L(A^*B) \subseteq L(X)$
      En effet : si X est une solution alors

$$\begin{array}{rcl} X & = & AX \cup B \\ X & = & A(AX \cup B) \cup B \\ X & = & A^2X \cup AB \cup B \\ & \dots & \\ X & = & A^{n+1}X \cup A^nB \cup A^{n-1}B \cup \ldots \cup AB \cup B \; \forall n \geq 0 \end{array}$$

      X contient donc tous les A^nB, $\forall n \geq 0$ et donc $L(A^*B) \subseteq L(X)$
   2. $L(X) \subseteq L(A^*B)$

Preuve

1. A^*B est une solution de l'équation $X = AX \cup B$.
   En effet : $AA^*B \cup B = (AA^* \cup \{\varepsilon\})B = A^*B$
2. S'il existe une autre solution X alors elle est équivalente à A^*B $(L(X) = L(A^*B))$
   1. $L(A^*B) \subseteq L(X)$
   2. $L(X) \subseteq L(A^*B)$

Preuve

1. A^*B est une solution de l'équation $X = AX \cup B$.
   En effet : $AA^*B \cup B = (AA^* \cup \{\varepsilon\})B = A^*B$
2. S'il existe une autre solution X alors elle est équivalente à A^*B ($L(X) = L(A^*B)$)
   1. $L(A^*B) \subseteq L(X)$
   2. $L(X) \subseteq L(A^*B)$
      En effet : soit $w \in L(X)$, notons $n = |w|$
      On a $w \in A^{n+1}X \cup A^nB \cup A^{n-1}B \cup \ldots \cup AB \cup B$ Comme les mots de $A^{n+1}X$ sont de longueur supérieure à n (puisque $\varepsilon \notin A$) on a $w \in A^nB \cup A^{n-1}B \cup \ldots \cup AB \cup B \subseteq A^*B$

Résolution du système d'équations

On considère les langages réguliers A_i^j, B_i tels qu'aucun A_i^j ne contienne ε. Alors, le système d'équations :

$$
\begin{array}{rcl}
L_1 & = & A_1^1 L_1 \cup \ldots \cup A_n^1 L_n \cup B_1 \\
 & \ldots & \\
L_n & = & A_n^1 L_1 \cup \ldots \cup A_n^n L_n \cup B_n
\end{array}
$$

admet une unique solution qui est un n-uplet $(L_1, \ldots, L_n)$ de langages réguliers

Exemple

$$X_1 = bX_1 + aX_2 \quad (1)$$

$$X_2 = aX_2 + bX_1 + \varepsilon \quad (2)$$

Exemple

$$
\begin{aligned}
X_1 &= bX_1 + aX_2 && (1)\\
X_2 &= aX_2 + bX_1 + \varepsilon && (2)
\end{aligned}
$$

On résoud 2 :

Exemple

$$\begin{aligned} X_1 &= bX_1 + aX_2 && (1) \\ X_2 &= aX_2 + bX_1 + \varepsilon && (2) \end{aligned}$$

On résoud 2 :

$$\begin{aligned} X_2 &= aX_2 + bX_1 + \varepsilon \\ &= a^*(bX_1 + \varepsilon) \end{aligned}$$

Exemple

$$
\begin{aligned}
X_1 &= bX_1 + aX_2 & (1)\\
X_2 &= aX_2 + bX_1 + \varepsilon & (2)
\end{aligned}
$$

On résoud 2 :

$$
\begin{aligned}
X_2 &= aX_2 + bX_1 + \varepsilon\\
&= a^*(bX_1 + \varepsilon)
\end{aligned}
$$

On remplace X_2 dans 1 :

Exemple

$$\begin{aligned} X_1 &= bX_1 + aX_2 && (1) \\ X_2 &= aX_2 + bX_1 + \varepsilon && (2) \end{aligned}$$

On résoud 2 :

$$\begin{aligned} X_2 &= aX_2 + bX_1 + \varepsilon \\ &= a^*(bX_1 + \varepsilon) \end{aligned}$$

On remplace X_2 dans 1 :

$$\begin{aligned} X_1 &= bX_1 + a(a^*(bX_1 + \varepsilon)) \\ &= bX_1 + aa^*bX_1 + aa^* \\ &= (b + aa^*b)X_1 + aa^* \\ &= (b + aa^*b)^*aa^* \end{aligned}$$

Equivalence des deux solutions

$$(b + aa^*b)^*aa^* \stackrel{?}{\equiv} b^*a(a + bb^*a)^*$$

Equivalence des deux solutions

$$(b + aa^*b)^*aa^* \stackrel{?}{\equiv} b^*a(a + bb^*a)^*$$

$$\begin{array}{rcll}
(b + aa^*b)^*aa^* & \equiv & ((\varepsilon + aa^*)b)^*aa^* & \\
& \equiv & (a^*b)^*aa^* & \alpha\alpha^* \equiv \alpha^*\alpha \\
& \equiv & (a^*b)^*a^*a & (\alpha^*\beta)^*\alpha^* \equiv (\alpha + \beta)^* \\
& \equiv & (a + b)^*a &
\end{array}$$

Equivalence des deux solutions

$$(b + aa^*b)^*aa^* \stackrel{?}{\equiv} b^*a(a + bb^*a)^*$$

$$\begin{array}{rcll}
(b + aa^*b)^*aa^* & \equiv & ((\varepsilon + aa^*)b)^*aa^* & \\
& \equiv & (a^*b)^*aa^* & \alpha\alpha^* \equiv \alpha^*\alpha \\
& \equiv & (a^*b)^*a^*a & (\alpha^*\beta)^*\alpha^* \equiv (\alpha + \beta)^* \\
& \equiv & (a + b)^*a &
\end{array}$$

$$\begin{array}{rcll}
b^*a(a + bb^*a)^* & \equiv & b^*a((\varepsilon + bb^*)a)^* & \\
& \equiv & b^*a(b^*a)^* & \alpha\alpha^* \equiv \alpha^*\alpha \\
& \equiv & (b^*a)^*b^*a & (\alpha^*\beta)^*\alpha^* \equiv (\alpha + \beta)^* \\
& \equiv & (a + b)^*a &
\end{array}$$

Plan du cours

- Langages résiduels
- Automate des résiduels
- Théorème de Myhill-Nerode

Langages résiduels

- On appelle résiduel d'un langage $L \subseteq \Sigma^*$ par rapport à un mot $u \in \Sigma^*$, noté L/u, l'ensemble des mots de L qui commencent par u auxquels on a retiré ce préfixe u.
- Autrement dit :

$$L/u = \{v \in \Sigma^* \mid uv \in L\}$$

- Exemple : Si $L = aa(bb + c)^*$ alors :
  - $L/a = a(bb + c)^*$
  - $L/b = \emptyset$
  - $L/c = \emptyset$

Quelques propriétés des langages résiduels

Soient $X, Y \subseteq \Sigma^*$, $u, v \in \Sigma^*$ et $a \in \Sigma$, on a :

Quelques propriétés des langages résiduels

Soient $X, Y \subseteq \Sigma^*$, $u, v \in \Sigma^*$ et $a \in \Sigma$, on a :

1. $X/uv = (X/u)/v$

Quelques propriétés des langages résiduels

Soient $X, Y \subseteq \Sigma^*$, $u, v \in \Sigma^*$ et $a \in \Sigma$, on a :

1. $X/uv = (X/u)/v$
2. $(X \cup Y)/u = (X/u) \cup (Y/u)$

Quelques propriétés des langages résiduels

Soient $X, Y \subseteq \Sigma^*$, $u, v \in \Sigma^*$ et $a \in \Sigma$, on a :

1. $X/uv = (X/u)/v$
2. $(X \cup Y)/u = (X/u) \cup (Y/u)$
3. si $\varepsilon \notin X$ alors $(XY)/a = (X/a)Y$

Quelques propriétés des langages résiduels

Soient $X, Y \subseteq \Sigma^*$, $u, v \in \Sigma^*$ et $a \in \Sigma$, on a :

1. $X/uv = (X/u)/v$
2. $(X \cup Y)/u = (X/u) \cup (Y/u)$
3. si $\varepsilon \notin X$ alors $(XY)/a = (X/a)Y$
4. si $\varepsilon \in X$ alors $(XY)/a = (X/a)Y \cup Y/a$

Quelques propriétés des langages résiduels

Soient $X, Y \subseteq \Sigma^*$, $u, v \in \Sigma^*$ et $a \in \Sigma$, on a :

1. $X/uv = (X/u)/v$
2. $(X \cup Y)/u = (X/u) \cup (Y/u)$
3. si $\varepsilon \notin X$ alors $(XY)/a = (X/a)Y$
4. si $\varepsilon \in X$ alors $(XY)/a = (X/a)Y \cup Y/a$
5. $\forall n > 0, X^n/a = (X/a)X^{n-1}$

Quelques propriétés des langages résiduels

Soient $X, Y \subseteq \Sigma^*$, $u, v \in \Sigma^*$ et $a \in \Sigma$, on a :

1. $X/uv = (X/u)/v$
2. $(X \cup Y)/u = (X/u) \cup (Y/u)$
3. si $\varepsilon \notin X$ alors $(XY)/a = (X/a)Y$
4. si $\varepsilon \in X$ alors $(XY)/a = (X/a)Y \cup Y/a$
5. $\forall n > 0, X^n/a = (X/a)X^{n-1}$
6. $X^*/a = (X/a)X^*$

Résiduels d'un langage décrit par une expression régulière

Soient X, Y des expressions régulières sur Σ, $u, v \in \Sigma^*$ et $a, b \in \Sigma$. On a :

Résiduels d'un langage décrit par une expression régulière

Soient X, Y des expressions régulières sur Σ, $u, v \in \Sigma^*$ et $a, b \in \Sigma$. On a :

1. $a/a = \varepsilon$

Résiduels d'un langage décrit par une expression régulière

Soient X, Y des expressions régulières sur Σ, $u, v \in \Sigma^*$ et $a, b \in \Sigma$. On a :

1. $a/a = \varepsilon$
2. $a/b = \emptyset$

Résiduels d'un langage décrit par une expression régulière

Soient X, Y des expressions régulières sur Σ, $u, v \in \Sigma^*$ et $a, b \in \Sigma$. On a :

1. $a/a = \varepsilon$
2. $a/b = \emptyset$
3. $(X + Y)/u = (X/u) + (Y/u)$

Résiduels d'un langage décrit par une expression régulière

Soient X, Y des expressions régulières sur Σ, $u, v \in \Sigma^*$ et $a, b \in \Sigma$. On a :

1. $a/a = \varepsilon$
2. $a/b = \emptyset$
3. $(X + Y)/u = (X/u) + (Y/u)$
4. $(XY)/a = (X/a)Y$ si $\varepsilon \notin L(X)$

Résiduels d'un langage décrit par une expression régulière

Soient X, Y des expressions régulières sur Σ, $u, v \in \Sigma^*$ et $a, b \in \Sigma$. On a :

1. $a/a = \varepsilon$
2. $a/b = \emptyset$
3. $(X + Y)/u = (X/u) + (Y/u)$
4. $(XY)/a = (X/a)Y$ si $\varepsilon \notin L(X)$
5. $(XY)/a = (X/a)Y + Y/a$ si $\varepsilon \in L(X)$

Résiduels d'un langage décrit par une expression régulière

Soient X, Y des expressions régulières sur Σ, $u, v \in \Sigma^*$ et $a, b \in \Sigma$. On a :

1. $a/a = \varepsilon$
2. $a/b = \emptyset$
3. $(X + Y)/u = (X/u) + (Y/u)$
4. $(XY)/a = (X/a)Y$ si $\varepsilon \notin L(X)$
5. $(XY)/a = (X/a)Y + Y/a$ si $\varepsilon \in L(X)$
6. $X^*/a = (X/a)X^*$

Résiduels d'un langage décrit par une expression régulière

Soient X, Y des expressions régulières sur Σ, $u, v \in \Sigma^*$ et $a, b \in \Sigma$. On a :

1. $a/a = \varepsilon$
2. $a/b = \emptyset$
3. $(X + Y)/u = (X/u) + (Y/u)$
4. $(XY)/a = (X/a)Y$ si $\varepsilon \notin L(X)$
5. $(XY)/a = (X/a)Y + Y/a$ si $\varepsilon \in L(X)$
6. $X^*/a = (X/a)X^*$
7. $X/uv = (X/u)/v$

Exemple

$(a+b)^*aba/ab$

$X/uv = (X/u)/v$

Exemple

$$\begin{array}{ll} & (a+b)^*aba/ab \\ = & ((a+b)^*aba/a)/b \end{array}$$

$X/uv = (X/u)/v$

$(XY)/a = (X/a)Y + Y/a$ si $\varepsilon \in L(X$

Exemple

$$
\begin{array}{lll}
 & (a+b)^*aba/ab & X/uv = (X/u)/v \\
= & ((a+b)^*aba/a)/b & (XY)/a = (X/a)Y + Y/a \text{ si } \varepsilon \in L(X \\
= & (((a+b)^*/a)aba + aba/a)/b & (XY)/a = (X/a)Y \text{ si } \varepsilon \notin L(X)
\end{array}
$$

Exemple

$$
\begin{array}{lll}
 & (a+b)^*aba/ab & X/uv = (X/u)/v \\
= & ((a+b)^*aba/a)/b & (XY)/a = (X/a)Y + Y/a \text{ si } \varepsilon \in L(X \\
= & (((a+b)^*/a)aba + aba/a)/b & (XY)/a = (X/a)Y \text{ si } \varepsilon \notin L(X) \\
= & (((a+b)^*/a)aba + (a/a)ba)/b & X^*/a = (X/a)(^*X)
\end{array}
$$

Exemple

$$
\begin{array}{rll}
 & (a+b)^*aba/ab & X/uv = (X/u)/v \\
= & ((a+b)^*aba/a)/b & (XY)/a = (X/a)Y + Y/a \text{ si } \varepsilon \in L(X \\
= & (((a+b)^*/a)aba + aba/a)/b & (XY)/a = (X/a)Y \text{ si } \varepsilon \notin L(X) \\
= & (((a+b)^*/a)aba + (a/a)ba)/b & X^*/a = (X/a)(^*X) \\
= & ((a+b)/a(a+b)^*aba + \varepsilon ba)/b & (X+Y)/u = (X/u) + (Y/u)
\end{array}
$$

Exemple

$$
\begin{array}{rll}
 & (a+b)^*aba/ab & X/uv = (X/u)/v \\
= & ((a+b)^*aba/a)/b & (XY)/a = (X/a)Y + Y/a \text{ si } \varepsilon \in L(X \\
= & (((a+b)^*/a)aba + aba/a)/b & (XY)/a = (X/a)Y \text{ si } \varepsilon \notin L(X) \\
= & (((a+b)^*/a)aba + (a/a)ba)/b & X^*/a = (X/a)(^*X) \\
= & ((a+b)/a(a+b)^*aba + \varepsilon ba)/b & (X+Y)/u = (X/u) + (Y/u) \\
= & ((a/a+b/a)(a+b)^*aba + ba)/b &
\end{array}
$$

Exemple

$$
\begin{array}{rll}
 & (a+b)^*aba/ab & X/uv = (X/u)/v \\
= & ((a+b)^*aba/a)/b & (XY)/a = (X/a)Y + Y/a \text{ si } \varepsilon \in L(X \\
= & (((a+b)^*/a)aba + aba/a)/b & (XY)/a = (X/a)Y \text{ si } \varepsilon \notin L(X) \\
= & (((a+b)^*/a)aba + (a/a)ba)/b & X^*/a = (X/a)(^*X) \\
= & ((a+b)/a(a+b)^*aba + \varepsilon ba)/b & (X+Y)/u = (X/u) + (Y/u) \\
= & ((a/a + b/a)(a+b)^*aba + ba)/b & \\
= & ((\varepsilon + \emptyset)(a+b)^*aba + ba)/b &
\end{array}
$$

Exemple

$$
\begin{array}{rll}
 & (a+b)^*aba/ab & X/uv = (X/u)/v \\
= & ((a+b)^*aba/a)/b & (XY)/a = (X/a)Y + Y/a \text{ si } \varepsilon \in L(X \\
= & (((a+b)^*/a)aba + aba/a)/b & (XY)/a = (X/a)Y \text{ si } \varepsilon \notin L(X) \\
= & (((a+b)^*/a)aba + (a/a)ba)/b & X^*/a = (X/a)({}^*X) \\
= & ((a+b)/a(a+b)^*aba + \varepsilon ba)/b & (X+Y)/u = (X/u) + (Y/u) \\
= & ((a/a + b/a)(a+b)^*aba + ba)/b & \\
= & ((\varepsilon + \emptyset)(a+b)^*aba + ba)/b & \\
= & ((a+b)^*aba + ba)/b &
\end{array}
$$

Exemple

$$
\begin{array}{cll}
 & (a+b)^*aba/ab & X/uv = (X/u)/v \\
= & ((a+b)^*aba/a)/b & (XY)/a = (X/a)Y + Y/a \text{ si } \varepsilon \in L(X \\
= & (((a+b)^*/a)aba + aba/a)/b & (XY)/a = (X/a)Y \text{ si } \varepsilon \notin L(X) \\
= & (((a+b)^*/a)aba + (a/a)ba)/b & X^*/a = (X/a)(^*X) \\
= & ((a+b)/a(a+b)^*aba + \varepsilon ba)/b & (X+Y)/u = (X/u) + (Y/u) \\
= & ((a/a+b/a)(a+b)^*aba + ba)/b & \\
= & ((\varepsilon + \emptyset)(a+b)^*aba + ba)/b & \\
= & ((a+b)^*aba + ba)/b & \\
= & (a+b)^*aba/b + ba/b &
\end{array}
$$

Exemple

$$
\begin{array}{lll}
 & (a+b)^*aba/ab & X/uv = (X/u)/v \\
= & ((a+b)^*aba/a)/b & (XY)/a = (X/a)Y + Y/a \text{ si } \varepsilon \in L(X \\
= & (((a+b)^*/a)aba + aba/a)/b & (XY)/a = (X/a)Y \text{ si } \varepsilon \notin L(X) \\
= & (((a+b)^*/a)aba + (a/a)ba)/b & X^*/a = (X/a)(^*X) \\
= & ((a+b)/a(a+b)^*aba + \varepsilon ba)/b & (X+Y)/u = (X/u) + (Y/u) \\
= & ((a/a+b/a)(a+b)^*aba + ba)/b & \\
= & ((\varepsilon + \emptyset)(a+b)^*aba + ba)/b & \\
= & ((a+b)^*aba + ba)/b & \\
= & (a+b)^*aba/b + ba/b & \\
= & ((a+b)^*/b)aba + aba/b + ba/b &
\end{array}
$$

Exemple

$$
\begin{array}{rll}
 & (a+b)^*aba/ab & X/uv = (X/u)/v \\
= & ((a+b)^*aba/a)/b & (XY)/a = (X/a)Y + Y/a \text{ si } \varepsilon \in L(X \\
= & (((a+b)^*/a)aba + aba/a)/b & (XY)/a = (X/a)Y \text{ si } \varepsilon \notin L(X) \\
= & (((a+b)^*/a)aba + (a/a)ba)/b & X^*/a = (X/a)(^*X) \\
= & ((a+b)/a(a+b)^*aba + \varepsilon ba)/b & (X+Y)/u = (X/u) + (Y/u) \\
= & ((a/a + b/a)(a+b)^*aba + ba)/b & \\
= & ((\varepsilon + \emptyset)(a+b)^*aba + ba)/b & \\
= & ((a+b)^*aba + ba)/b & \\
= & (a+b)^*aba/b + ba/b & \\
= & ((a+b)^*/b)aba + aba/b + ba/b & \\
= & (a+b)^*aba + a &
\end{array}
$$

Langages L_q

- Pour chaque état q d'un automate $A = \langle Q, \Sigma, \delta, q_0, F \rangle$, on note L_q l'ensemble des mots w tels que le calcul de A sur w à partir de q aboutit à un état d'acceptation.
- Autrement dit :

$$L_q = \{ w \in \Sigma^* \mid (q, w) \overset{*}{\vdash} (q_f, \varepsilon) \text{ avec } q_f \in F \}$$

- En particulier : $L_{q_0} = L(A)$

Langages L_q et résiduels

- Soient $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ un automate déterministe et $L = L(A)$.
- $\forall u \in \Sigma^*$ et $\forall q \in Q$, on a :

$$\text{si } (q_0, u) \overset{*}{\vdash} (q, \varepsilon) \text{ alors } L/u = L_q$$

Langages L_q et résiduels

- Soient $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ un automate déterministe et $L = L(A)$.
- $\forall u \in \Sigma^*$ et $\forall q \in Q$, on a :

$$\text{si } (q_0, u) \overset{*}{\vdash} (q, \varepsilon) \text{ alors } L/u = L_q$$

- Preuve

$$\begin{aligned}
x \in L/u \quad & \Leftrightarrow \quad ux \in L \\
& \Leftrightarrow \quad (q_0, ux) \overset{*}{\vdash} (q_f, \varepsilon) \text{ avec } q_f \in F \\
& \Leftrightarrow \quad (q_0, ux) \overset{*}{\vdash} (q, x) \overset{*}{\vdash} (q_f, \varepsilon) \\
& \Leftrightarrow \quad (q, x) \overset{*}{\vdash} (q_f, \varepsilon) \\
& \Leftrightarrow \quad x \in L_q
\end{aligned}$$

Langages L_q et résiduels

- L'ensemble des résiduels d'un langage reconnaissable est égal à ses langages L_q
- Autrement dit, si $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ est un automate déterministe complet. Alors :

$$\{L(A)/w \mid w \in \Sigma^*\} = \{L_q \mid q \in Q\}$$

Par conséquent

- Le nombre de résiduels de $L(A)$ est fini
- le nombre de résiduels de $L(A)$ est inférieur ou égal à $|Q|$ car il peut exister deux états $q, q' \in Q$ tels que $L_q = L_{q'}$.

Automate des résiduels

- Lorsqu'un langage n'a qu'un nombre fini de résiduels, on peut lui associer un automate particulier appelé automate des résiduels de L
- Soit $L \subseteq \Sigma^*$ un langage tel que $L/w, w \in \Sigma^*$ est fini.
- L'automate des résiduels de L est l'automate déterministe $A_L = \langle Q, \Sigma, \delta, q_0, F \rangle$ défini par :
  - $Q = \{L/w, w \in \Sigma^*\}$
  - $\delta(L/w, a) = L/wa$ pour tout $w \in \Sigma^*, a \in \Sigma$
  - $q_0 = L$
  - $F = \{L/w, w \in L\}$
- L'automate des résiduels de L reconnaît L

Exemple

Les résiduels du langage L dénoté par l'expression régulière $ab^* + ba^*$ sont :

- $L = L/\varepsilon$
- $L/a = L/ab = b^*$
- $L/b = L/ba = a^*$
- $L/aa = L/bb = \emptyset$

Ce qui donne pour L l'automate des résiduels suivant :

Théorème de Myhill-Nerode

- Un langage est reconnaissable si et seulement s'il n'a qu'un nombre fini de résiduels

Théorème de Myhill-Nerode

- Un langage est reconnaissable si et seulement s'il n'a qu'un nombre fini de résiduels

Preuve :

- Un langage reconnaissable a un nombre fini de résiduels
- Réciproquement, pour tout langage L comportant un nombre fini de résiduels, on peut construire son automate des résiduels