Théorie des Langages 1

Cours 6 : Expressions régulières

L. Rieg

Grenoble INP - Ensimag, 1re année

Année 2024-2025

Définition

Définition (Expression régulière)

L'ensemble des expressions régulières (sur un vocabulaire V) est défini par induction structurelle par :

- **Base :**
  - $\emptyset$ est une expression régulière
  - ϵ est une expression régulière
  - Si $x \in V$, alors x est une expression régulière
- **Induction** : Si E_1 et E_2 sont des expressions régulières, alors
  - $(E_1 + E_2)$ est une expression régulière
  - $(E_1.E_2)$ est une expression régulière
  - $({E_1}^*)$ est une expression régulière

Exemple : $(a.((a + b)^*))$ est une expression régulière (ER).

Abréviations

- On pourra noter E_1E_2 à la place de $E_1.E_2$
- On pourra supprimer les parenthèses « inutiles » en éliminant la paire la plus externe et en considérant
  - « . » et « + » associatifs
  - « * » plus prioritaire que « . » plus prioritaire que « + »

Exemple

Soit $E = ((a.(b + (c + (d.(c^*))))).((a.b)^*))$.
On pourra simplement noter $E = a(b + c + dc^*)(ab)^*$.

Question : à quoi servent les ER ?

- Intérêt pratique : recherche de motif (cf. `grep` et `sed`)
- Intérêt théorique : description inductive des langages réguliers

Langage représenté

Une expression régulière sur V est un mot sur $V \cup \{\emptyset, \epsilon,), (, +, ^*, .\}$ qui représente un langage sur V.

Définition

Le langage représenté par une ER E est noté $\mathcal{L}(E)$ et est défini par :

- Si $E = \emptyset$ alors $\mathcal{L}(E) =$
- Si $E = \epsilon$ alors $\mathcal{L}(E) =$
- Si $E = x \quad (x \in V)$ alors $\mathcal{L}(E) =$
- Si $E = (E_1 + E_2)$ alors $\mathcal{L}(E) =$
- Si $E = (E_1.E_2)$ alors $\mathcal{L}(E) =$
- Si $E = ({E_1}^*)$ alors $\mathcal{L}(E) =$

Remarques

Exemple

$\mathcal{L}((a+ab)^*) \quad =$

Remarques

- *L'associativité dans les ER (pour « . » et « + ») vient des langages.*
- *Priorités comme en arithmétique : « * » > « . » > « + »*

Notations

- *On pourra noter E à la place de $\mathcal{L}(E)$.*
- *Du coup, des notations comme $w \in E$ ou $A \subseteq B + C$ sont autorisées.*
- *Il ne faut pas tout mélanger : on n'écrira pas $w \in (a+b)^* \{c, d\}$.*

Expressions régulières équivalentes

Définition

Deux expressions régulières E et E' sont équivalentes si $\mathcal{L}(E) = \mathcal{L}(E')$.

Exemple

Les expressions régulières $(a+b)^*$ et $(a^*b^*)^*$ sont équivalentes.
Exercice : démontrer ce résultat

Théorème (Kleene)

Les langages représentés par des expressions régulières sont les langages réguliers.

Rappel : L régulier $\stackrel{\text{def}}{=}$ $\exists A(\text{AF}) : \mathcal{L}(A) = L$

À démontrer :
1. $\forall E(\text{ER}), \exists A(\text{AF}) : \mathcal{L}(E) = \mathcal{L}(A)$
2. $\forall A(\text{AF}), \exists E(\text{ER}) : \mathcal{L}(A) = \mathcal{L}(E)$

Une ER représente un langage régulier

Lemme

$\forall E(ER), \exists A(AF)$ tel que $\mathcal{L}(E) = \mathcal{L}(A)$
De plus, A a un unique état initial et un unique état acceptant.

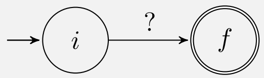
Preuve : par induction structurelle

- $E = \emptyset$
- $E = \epsilon$
- $E = x \in V$
- $E = (E_1 + E_2),\ (E_1.E_2),\ (E_1^*)$

Un langage régulier est représentable par une ER

2 visions possibles :

- Version graphique :
  - Idée : se ramener à un automate de la forme



où ? est une ER

  - Méthode :
    - S'autoriser des ER sur les transitions
    - Partir d'un automate avec un unique état initial i sans transition entrante et un unique état acceptant f sans transition sortante (voir cours 3)
    - Supprimer successivement les états (sauf i et f) en préservant le langage reconnu
- Version par équations : résolution d'un système d'équations d'ER
  - Pas besoin de transformer l'automate
  - Représentation algébrique de la version graphique

Version graphique

Comment supprimer un état d'un automate sans changer son langage ?

Formellement, pour supprimer un état q avec

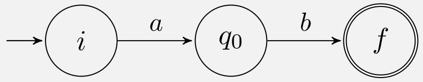
- des transitions entrantes $(p, x, q) \in \delta$ (avec $p \neq q$)
- possiblement une boucle $(q, y, q) \in \delta$
- des transitions sortantes $(q, z, r) \in \delta$ (avec $r \neq q$)

on doit

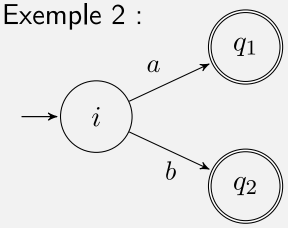
C'est la méthode de Brzozowski et Mc Cuskey.
On peut programmer cette méthode : l'algorithme de Kleene (prog. dyn.)

Version graphique : exemples

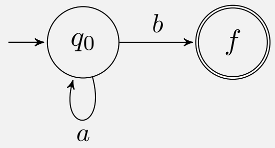
- Exemple 1 :



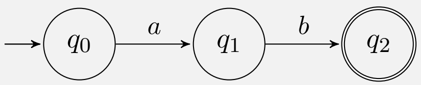
- Exemple 2 :



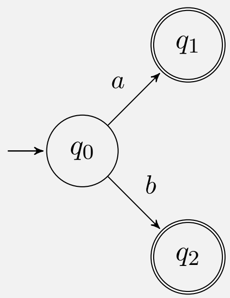
- Exemple 3 :



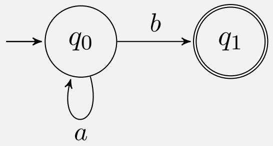
Version par équations : exemple 1



Version par équations : exemple 2



Version par équations : exemple 3



Système d'équations associé à un automate A

Pour chaque état q_i :

- On considère une variable x_i, qui représentera le langage reconnu par l'automate dont q_i est l'unique état initial (A_{q_i})
- On considère les k transitions issues de q_i
  - $(q_i, a_{i_1}, q_{i_1}), (q_i, a_{i_2}, q_{i_2}), \ldots, (q_i, a_{i_k}, q_{i_k})$ où $a_{i_j} \in V \cup \{\varepsilon\}$
  - On crée l'équation :

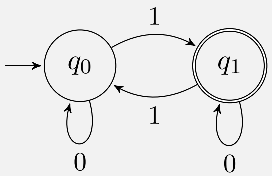
$$x_i = \sum_{j=1}^{k} a_{i_j} x_{i_j} \quad + \epsilon \text{ si } q_i \text{ est final}$$

Remarque : Si $k = 0$, la somme se réduit à $\emptyset$ et on a alors
$x_i = \emptyset$ si $q_i \notin F$ ou $x_i = \epsilon$ si $q_i \in F$

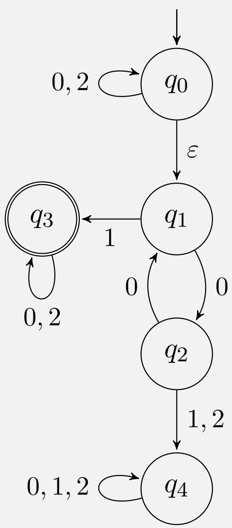
Théorème (Admis)

Pour tout i, la plus petite solution de l'équation associée à q_i représente le langage reconnu par l'automate dont q_i est l'unique état initial.

Exemple



Exercice



Résolution des systèmes d'équations

Lemme d'Arden

Soient A et B des langages, et considérons l'équation $X = AX + B$.
Alors :

- A^*B est la plus petite solution de cette équation
- Si $\varepsilon \notin A$, c'est l'unique solution

Preuve

- A^*B est une solution

$$\begin{aligned} A(A^*B) + B &= \\ &= \\ &= \\ &= A^*B \end{aligned}$$

Résolution des systèmes d'équations

Preuve (suite)

- A^*B est la plus petite solution

  Soit C une solution : on a $C = AC + B$.
  Donc $B \subseteq C$ et $AC \subseteq C$
  Donc $AB \subseteq AC \subseteq C$

  $\vdots$ Preuve par récurrence sur k (exercice)

  Donc $\forall k \geq 0,\ A^kB \subseteq C$

  Ainsi $A^*B \subseteq C$

Résolution des systèmes d'équations

Preuve (suite)

- Si $\varepsilon \notin A$ alors A^*B est l'unique solution.

  Soit C une solution, montrons que $C = A^*B$.
  Comme A^*B est la plus petite solution, on a $A^*B \subseteq C$. Il suffit donc de montrer que $C \subseteq A^*B$.
  Par l'absurde : on suppose que $\exists w \in C$ tel que $w \notin A^*B$.
  Supposons w de longueur minimale

Remarques importantes

Questions

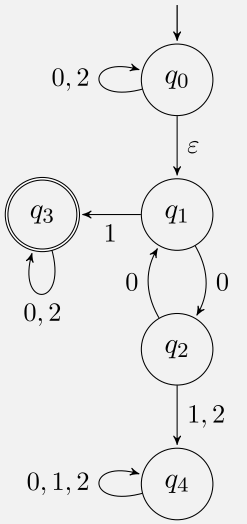
- Quelle est la (plus petite) solution de l'équation $X = AX + \epsilon$?
- Quelle est la (plus petite) solution de l'équation $X = AX$?
- Si l'automate a deux états initiaux q_j et q_k ?
- Si $\varepsilon \in A$, quelles autres solutions de $X = AX + B$ y a-t-il ?
- Qu'obtient-on pour l'équation $X = XA + B$?

Exercice

Résoudre le système d'équations suivant (q_0 état initial) :

$$\begin{cases} x_0 = (0+2)x_0 + x_1 \\ x_1 = 0x_2 + 1x_3 \\ x_2 = 0x_1 + (1+2)x_4 \\ x_3 = (0+2)x_3 + \epsilon \\ x_4 = (0+1+2)x_4 \end{cases}$$

« Vérification » a posteriori

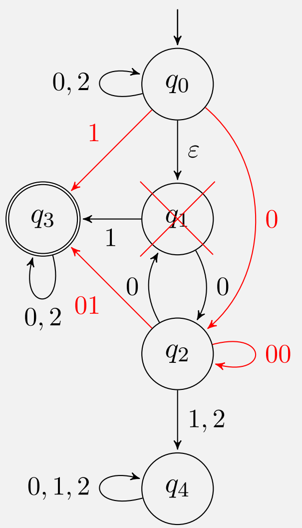


On peut s'assurer que l'automate « reconnaît » l'expression régulière calculée.

Remarque : on peut trouver différentes expressions régulières en fonction de l'ordre dans lequel les équations sont résolues.

On peut faire correspondre la substitution d'une variable et la suppression de l'état correspondant dans la méthode graphique.

Correspondance entre versions graphique et par équation



Avant suppression :

$$\begin{cases} x_0 = (0+2)x_0 + x_1 \\ x_1 = 0x_2 + 1x_3 \\ x_2 = 0x_1 + (1+2)x_4 \\ x_3 = (0+2)x_3 + \epsilon \\ x_4 = (0+1+2)x_4 \end{cases}$$

Après suppression de q_1 :

$$\begin{cases} x_0 = (0+2)x_0 + 0x_2 + 1x_3 \\ x_2 = 0(0x_2 + 1x_3) + (1+2)x_4 \\ x_3 = (0+2)x_3 + \epsilon \\ x_4 = (0+1+2)x_4 \end{cases}$$