Langages et Automates : LA3
Partie 4 : De l'Automate à l'Expression Rationnelle

De l'automate à l'expression rationnelle

On va voir deux algorithmes (et donc deux preuves de Rec $\subset$ Rac) pour réaliser ceci.

Systémes d'équations

Soit $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$ un automate. On définit les langages :

$$\forall q \in Q,\ L_q = \{w \in \Sigma^*,\ \delta^*(q, w) \in F\}$$

cad. l'ensemble des mots qui sont acceptés "à partir de l'état q".

On a alors le système d'équations (linéaires gauches)

$$\begin{cases} L_q = \sum\limits_{a \in \Sigma} a.L_{\delta(q,a)} & \forall q \in Q \setminus F \\ L_q = \sum\limits_{a \in \Sigma} a.L_{\delta(q,a)} + \varepsilon & \forall q \in F \end{cases}$$

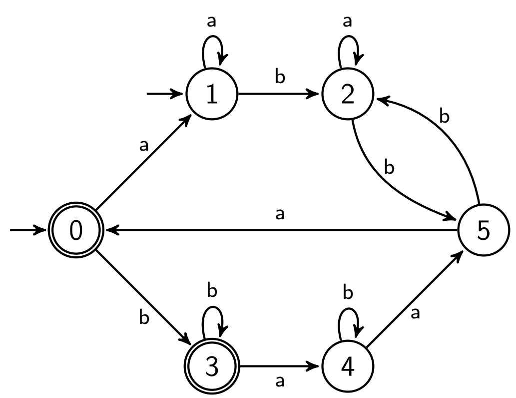
Résolution - Lemme d'Arden

On peut résoudre un tel système grâce au résultat suivant :

Théoreme (Lemme d'Arden)

Une équation de la forme $L = A.L + B$ où A ne contient pas le mot vide admet comme unique solution l'expression rationnelle $L = A^*.B$

Exemple :



Pour cet algorithme, on va étendre la notion d'automate : automate généralisé.
Les transitions peuvent maintenant être étiquetées par une expression rationnelle.
Pour un tel automate un mot w est accepté si il existe une décomposition $w = w_1 ... w_n$ et un ensemble d'états $q_0, ..., q_n$ tels que

- q_0 est un état initial de l'automate
- q_n est un état acceptant.
- pour tout i, il existe un état q_i et une transition de q_{i-1} vers q_i étiquetée par une E.R. à laquelle le mot w_i appartient.

L'idée de l'algo est alors de supprimer un à un les états de l'automate afin d'arriver à un automate équivalent ne contenant plus que deux états, un initial, un acceptant, l'expression qui étiquette la transition entre les deux est alors l'expression recherchée.

On suppose que les états de $\mathcal{A}$ sont $\{q_1, \ldots, q_n\}$. On commence par ajouter un état initial q_0 et un état q_{n+1}. On ajoute une ε-transition de q_0 vers tous les états initiaux de $\mathcal{A}$. De même, on ajoute une ε-transition de tous les états acceptants vers q_{n+1}.

A un moment donné de l'algorithme, on notera E_{ij}, l'expression qui étiquette la transition de l'état q_i vers l'état q_j (si aucune transition existe, cela est équivalent à dire que $E_{ij} = \emptyset$).

L'algorithme consiste alors à supprimer successivement $q_1, \ldots, q_n$ en appliquant la regle suivante lorsque l'on supprime q_i :
Pour touts k, l différents de i, E_{kl} est remplacée par $E_{kl} + E_{ki}(E_{ii})^* Eij$.