Les expressions rationnelles et les langages rationnels

Les langages rationnels

On appelle ensemble des langages rationnels sur Σ et on note $Rat(\Sigma^*)$ le plus petit ensemble contenant $\emptyset$, $\{a\}$, $\forall a \in \Sigma$ et fermé pour l'union, le produit de concaténation et l'étoile.

exemple :
Soit $L = \{w \in \Sigma^* \text{ tq } |w|_a mod2 = 0\} = (\{b\}^* \cdot \{a\} \cdot \{b\}^* \cdot \{a\})^* \cdot \{b\}^* \in Rat(\Sigma^*)$.

On rappelle que $\emptyset^* = \{\varepsilon\}$.

Théorème de Kleene :

$$Rec(\Sigma^*) = Rat(\Sigma^*)$$

Les expressions rationnelles

— $\emptyset$ est une expression rationnelle qui désigne le langage rationnel $\emptyset$
— ε est une expression rationnelle qui désigne le langage rationnel $\{\varepsilon\}$
— $a \in \Sigma$ est une expression rationnelle qui désigne le langage rationnel $\{a\}$

si r et s sont deux expressions rationnelles qui désignent respectivement les langages rationnels R et S, alors

— $(r + s)$ est une expression rationnelle qui désigne le langage rationnel $R + S$
— $(r.s)$ est une expression rationnelle qui désigne le langage rationnel $R.S$
— (r^*) est une expression rationnelle qui désigne le langage rationnel R^*

remarque : il ne faut pas confondre ε expression rationnelle qui désigne le langage $\{\varepsilon\}$ et ε le mot vide.

Une expression rationnelle désigne exactement un langage rationnel et un langage rationnel est désigné par au moins une expression rationnelle.

exemple : $(a^* + b^*)^*$, $(a^* \cdot b^*)^*$ et $(a + b)^*$ désignent le même langage rationnel Σ^*.

La démonstration du théorème de Kleene repose sur deux algorithmes ; l'un permet de passer d'une expression rationnelle à un automate et le deuxième d'un automate à une expression rationnelle.

Construction d'un automate à partir d'une expression rationnelle

Nous présentons ici le détail d'une méthode de construction : l'algorithme de Glushkov.

L'algorithme de Glushkov

L'algorithme de Glushkov prend en entrée une expression rationnelle et retourne un automate de Glushkov.

Soit E une expression rationnelle sur l'alphabet Σ et $L(E)$ le langage désigné par cette expression. On numérote chaque lettre de E de la gauche vers la droite en commençant à 1. Chaque numéro de lettre est appelé position de cette lettre dans E et on note $Pos(E)$ l'ensemble des positions des lettres de E. Soit $\overline{E}$ la nouvelle expression ainsi indicée et $L(\overline{E})$ le langage désigné par cette expression. Notons a_i la lettre qui apparait en position i dans $\overline{E}$. Chaque mot w de $L(\overline{E})$ commence par une lettre a_i. La position i est appelée position initiale de w. De la même manière, chaque mot w de $L(\overline{E})$ se termine par une lettre a_j. La position j est appelée position finale de w.

Exemple 2 : soit $E = (a \cdot b^* + a) \cdot b$.
L'expression indicée correspondante est $\overline{E} = (a_1 \cdot b_2^* + a_3) \cdot b_4$
$Pos(E) = \{1, 2, 3, 4\}$.

On définit les fonctions suivantes :

- $First(E)$ est l'ensemble des positions initiales des mots de $L(\overline{E})$.
- $Last(E)$ est l'ensemble des positions finales des mots de $L(\overline{E})$.
- $Follow(E, i)$ est l'ensemble des positions des lettres qui suivent a_i dans les mots de $L(\overline{E})$.
- $Null(E) = \varepsilon$ si ε est dans $L(E)$ et $\emptyset$ sinon.

Exemple 3 :

- $\overline{E} = (a_1 \cdot b_2^* + a_3) \cdot b_4$
- $First(E) = \{1, 3\}$
- $Last(E) = \{4\}$
- $Null(E) = \emptyset$
- $Follow(E, 1) = \{2, 4\}$, $Follow(E, 2) = \{2, 4\}$, $Follow(E, 3) = \{4\}$, $Follow(E, 4) = \emptyset$.

Calcul des fonctions et des ensembles de positions

rappels :
$\emptyset \cdot E = E \cdot \emptyset = \emptyset$, $\forall E$ une expression rationnelle
$\emptyset^* = \varepsilon$

La fonction *Null*

$$
\begin{array}{lll}
Null(\emptyset) & = & \emptyset \\
Null(\varepsilon) & = & \{\varepsilon\} \\
Null(a) & = & \emptyset \\
Null(F+G) & = & Null(F) \cup Null(G) \\
Null(F\cdot G) & = & Null(F) \cap Null(G) \\
Null(F^*) & = & \{\varepsilon\}
\end{array}
$$

La fonction *First*

$$
\begin{array}{llll}
First(\emptyset) & = & \emptyset & \\
First(\varepsilon) & = & \emptyset & \\
First(a) & = & \{x\} & \text{si } \overline{a} = \alpha_x \\
First(F+G) & = & First(F) \cup First(G) & \\
First(F\cdot G) & = & First(F) \cup Null(F)\cdot First(G) & \\
First(F^*) & = & First(F) &
\end{array}
$$

La fonction *Last*

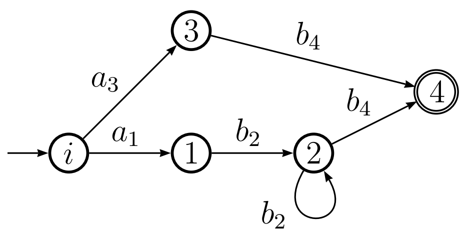
$$
\begin{array}{llll}
Last(\emptyset) & = & \emptyset & \\
Last(\varepsilon) & = & \emptyset & \\
Last(a) & = & \{x\} & \text{si } \overline{a} = \alpha_x \\
Last(F+G) & = & Last(F) \cup Last(G) & \\
Last(F\cdot G) & = & Last(G) \cup Null(G)\cdot Last(F) & \\
Last(F^*) & = & Last(F) &
\end{array}
$$

La fonction *Follow*

$$
\begin{array}{lll}
Follow(\emptyset, x) & = & \emptyset \\
Follow(\varepsilon, x) & = & \emptyset \\
Follow(a, x) & = & \emptyset \\
Follow(F+G, x) & = & \mathcal{I}_{Pos(F)}(x)\cdot Follow(F,x) \cup \mathcal{I}_{Pos(G)}(x)\cdot Follow(G,x) \\
Follow(F\cdot G, x) & = & \mathcal{I}_{Pos(F)}(x)\cdot Follow(F,x) \\
 & & \cup \mathcal{I}_{Pos(G)}(x)\cdot Follow(G,x) \\
 & & \cup \mathcal{I}_{Last(F)}(x)\cdot First(G) \\
Follow(F^*, x) & = & Follow(F,x) \cup \mathcal{I}_{Last(F)}(x)\cdot First(F)
\end{array}
$$

Ces fonctions permettent de définir l'automate $\overline{M}$ qui reconnaît le langage $L(\overline{E})$. On a $\overline{M} = (\Sigma, Q, \{i\}, F, \delta)$

1. $Q = Pos(E) \cup \{i\}$
2. $\forall x \in First(E), \delta(i, \alpha_x) = \{x\}$
3. $\forall x \in Pos(E), \forall y \in Follow(E,x), \delta(x, \alpha_y) = \{y\}$
4. $F = Last(E) \cup Null(E)\cdot \{i\}$



On obtient M à partir de $\overline{M}$ en enlevant les indices des lettres de l'alphabet.

Construction d'une expression rationnelle associée à un automate : algorithme de Mac Naughton et Yamada

Soit $M = (\Sigma, Q, I, F, \delta)$ un automate avec $Q = \{1, 2, \dots n\}$.
On pose
$R_{ij} = \{w \in \Sigma^*$ tq w est la trace d'un chemin quelconque de l'état i à l'état j dans M$\}$

$R_{ij}^k = \{w \in R_{ij}$ tq les chemins de i à j d'étiquette w passent par des états intermédiaires inférieurs ou égaux à $k\}$

Considérons un chemin de i à j dont la trace est dans R_{ij}^k.

- Soit ce chemin ne passe pas par k. Comme les états intermédiaires doivent être inférieurs ou égaux à k, dans ce cas, ils sont strictement inférieurs à k et donc inférieurs ou égaux à $k-1$ et leurs traces sont dans R_{ij}^{k-1}.
- Soit ce chemin passe par l'état k (éventuellement plusieurs fois).

Ce chemin peut être décomposé de la manière suivante :

$$i \longrightarrow k \longrightarrow k \longrightarrow k \dots \longrightarrow k \longrightarrow j$$

en mettant en évidence chaque occurence de k sur le chemin de i à j.

Tous les chemins de i à k, de k à k et de k à j passent par des états intermédiaires inférieurs strictement à k et donc inférieurs ou égaux à $k-1$. Leurs traces sont donc respectivement dans R_{ik}^{k-1}, R_{kk}^{k-1} et R_{kj}^{k-1}.

On a donc une décomposition de R_{ij}^k en :

$$R_{ij}^k = R_{ij}^{k-1} \cup R_{ik}^{k-1} \cdot (R_{kk}^{k-1})^* \cdot R_{kj}^{k-1} \text{ pour } k > 0$$

Il faut maintenant définir R_{ij}^0. Ce sont les traces des chemins allant i à j en passant par des états intermédiaires inférieurs ou égaux à 0. Mais on a posé $Q = \{1, 2, ...n\}$ et il n'existe par conséquent aucun état inférieur ou égal à 0 dans Q. Cela signifie que ces chemins ne peuvent pas avoir d'état intermédiaire. Ce sont donc des flêches et R_{ij}^0 est l'ensemble des étiquettes de ces flêches :

$$R_{ij}^0 = \{a \in \Sigma \mid j \in \delta(i, a)\}$$

On peut maintenant revenir à R_{ij}. C'est l'ensemble des traces des chemins allant de i à j sans condition sur les états intermédiaires ni sur la longueur. Autrement dit , on peut passer par n'importe quel état intermédiaire c'est à dire par n'importe quel état inférieur ou égal à n et on peut donc écrire $R_{ij} = R_{ij}^n$

On peut alors définir le langage reconnu par l'automate par :

$$R = \begin{cases} \bigcup_{i \in I, t \in F} R_{it} & \text{si } I \cap F = \emptyset \\ \bigcup_{i \in I, t \in F} R_{it} \cup \{\varepsilon\} & \text{sinon} \end{cases}$$

En résumé, on a donc les formules suivantes pour les langages :

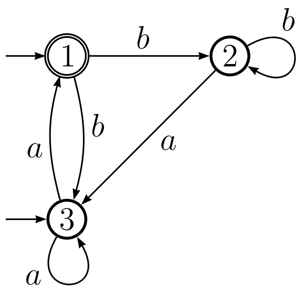
$$
\begin{array}{lcll}
R & = & \left\{\begin{array}{ll} \bigcup_{i\in I, t\in F} R_{it} & \text{si } I\cap F=\emptyset \\ \bigcup_{i\in I, t\in F} R_{it}\cup\{\varepsilon\} & \text{sinon} \end{array}\right. & \\
R_{ij} & = & R_{ij}^{n} & \\
R_{ij}^{k} & = & R_{ij}^{k-1}\cup R_{ik}^{k-1}\cdot (R_{kk}^{k-1})^{*}\cdot R_{kj}^{k-1} & \forall k>0 \\
R_{ij}^{0} & = & \{a\in\Sigma \mid j\in\delta(i,a)\} & \forall i,j\in Q
\end{array}
$$

On obtient alors les formules suivantes pour les expressions rationnelles qui désignent les langages ci-dessus :

$$
\begin{array}{lcll}
r & = & \left\{\begin{array}{ll} \displaystyle\sum_{i\in I, t\in T} r_{it} & \text{si } I\cap F=\emptyset \\ \displaystyle\sum_{i\in I, t\in T} r_{it}+\varepsilon & \text{sinon} \end{array}\right. & \\
r_{ij} & = & r_{ij}^{n}, & \forall i,j\in Q \\
r_{ij}^{k} & = & r_{ij}^{k-1}+r_{ik}^{k-1}\cdot (r_{kk}^{k-1})^{*}\cdot r_{kj}^{k-1} & \forall k>0 \\
r_{ij}^{0} & = & \displaystyle\sum_{a\in\Sigma, j\in\delta(i,a)} a, & \forall i,j\in Q
\end{array}
$$

Exemple :

On considère l'automate :



L'expression rationnelle obtenue à partir de cet automate par l'algorithme de McNaughton et Yamada est donnée par les formules :

$$r = r_{11} + r_{31} + \varepsilon = r_{11}^{3} + r_{31}^{3} + \varepsilon$$

Pour calculer ces expressions, il est nécessaire de calculer les r_{ij}^{k} pour $k = 0, 1, 2, 3$. Nous allons mettre ces calculs dans une table appelée table de McNaughton et Yamada :

r_{ij}	$k=0$	$k=1$	$k=1$	$k=2$	$k=2$	$k=3$
r_{11}		$r_{11}^0 + r_{11}^0 \cdot (r_{11}^0)^* \cdot r_{11}^0$		$r_{11}^1 + r_{12}^1 \cdot (r_{22}^1)^* \cdot r_{21}^1$		$r_{11}^2 + r_{13}^2 \cdot (r_{33}^2)^* \cdot r_{31}^2$
r_{12}		$r_{12}^0 + r_{11}^0 \cdot (r_{11}^0)^* \cdot r_{12}^0$		$r_{12}^1 + r_{12}^1 \cdot (r_{22}^1)^* \cdot r_{22}^1$		$r_{12}^2 + r_{13}^2 \cdot (r_{33}^2)^* \cdot r_{32}^2$
r_{13}		$r_{13}^0 + r_{11}^0 \cdot (r_{11}^0)^* \cdot r_{13}^0$		$r_{13}^1 + r_{12}^1 \cdot (r_{22}^1)^* \cdot r_{23}^1$		$r_{13}^2 + r_{13}^2 \cdot (r_{33}^2)^* \cdot r_{33}^2$
r_{21}		$r_{21}^0 + r_{21}^0 \cdot (r_{11}^0)^* \cdot r_{11}^0$		$r_{21}^1 + r_{22}^1 \cdot (r_{22}^1)^* \cdot r_{21}^1$		$r_{21}^2 + r_{23}^2 \cdot (r_{33}^2)^* \cdot r_{31}^2$
r_{22}		$r_{22}^0 + r_{21}^0 \cdot (r_{11}^0)^* \cdot r_{12}^0$		$r_{22}^1 + r_{22}^1 \cdot (r_{22}^1)^* \cdot r_{22}^1$		$r_{22}^2 + r_{23}^2 \cdot (r_{33}^2)^* \cdot r_{32}^2$
r_{23}		$r_{23}^0 + r_{21}^0 \cdot (r_{11}^0)^* \cdot r_{13}^0$		$r_{23}^1 + r_{22}^1 \cdot (r_{22}^1)^* \cdot r_{23}^1$		$r_{23}^2 + r_{23}^2 \cdot (r_{33}^2)^* \cdot r_{33}^2$
r_{31}		$r_{31}^0 + r_{31}^0 \cdot (r_{11}^0)^* \cdot r_{11}^0$		$r_{31}^1 + r_{32}^1 \cdot (r_{22}^1)^* \cdot r_{21}^1$		$r_{31}^2 + r_{33}^2 \cdot (r_{33}^2)^* \cdot r_{31}^2$
r_{32}		$r_{32}^0 + r_{31}^0 \cdot (r_{11}^0)^* \cdot r_{12}^0$		$r_{32}^1 + r_{32}^1 \cdot (r_{22}^1)^* \cdot r_{22}^1$		$r_{32}^2 + r_{33}^2 \cdot (r_{33}^2)^* \cdot r_{32}^2$
r_{33}		$r_{33}^0 + r_{31}^0 \cdot (r_{11}^0)^* \cdot r_{13}^0$		$r_{33}^1 + r_{32}^1 \cdot (r_{22}^1)^* \cdot r_{23}^1$		$r_{33}^2 + r_{33}^2 \cdot (r_{33}^2)^* \cdot r_{33}^2$

Dans notre exemple, on obtient la table complètée suivante :

r_{ij}	$k=0$	$k=1$	$k=2$
r_{11}	$\emptyset$	$\emptyset$	$\emptyset$
r_{12}	b	b	$b+b\cdot b^*\cdot b$
r_{13}	b	b	$b+b\cdot b^*\cdot a$
r_{21}	$\emptyset$	$\emptyset$	$\emptyset$
r_{22}	b	b	$b+b\cdot b^*\cdot b$
r_{23}	a	a	$a+b\cdot b^*\cdot a$
r_{31}	a	a	a
r_{32}	$\emptyset$	$a\cdot b$	$a\cdot b+a\cdot b\cdot b^*\cdot b$
r_{33}	a	$a+a\cdot b$	$a+a\cdot b+a\cdot b\cdot b^*\cdot a$

$$\text{Soit } r = \varepsilon + (b+b\cdot b^*\cdot a)\cdot(a+a\cdot b+a\cdot b\cdot b^*\cdot a)^*\cdot a$$
$$+\,(a+(a+a\cdot b+a\cdot b\cdot b^*\cdot a)\cdot(a+a\cdot b+a\cdot b\cdot b^*\cdot a)^*\cdot a)$$