Automates finis [AF3] - Notes de cours 4
Langages rationnels et expressions rationnelles
Passage d'un automate fini à une expression rationnelle
Le théorème de Kleene

Jean-Marie Rifflet - Octobre 2005

---

1. La famille $\mathcal{R}at(X^*)$ des langages rationnels

Nous nous intéressons ici à la famille $\mathcal{R}$at(X*) des **langages** appelés **rationnels** (aussi appelés **langages réguliers** ou **K-langages**) dont nous allons montrer (**théorème de Kleene**) qu'elle est identique à la famille $\mathcal{R}$ec(X*).

1.1. Définition

La famille $\mathcal{R}$at(X*) est la plus petite famille de langages satisfaisant les conditions suivantes :

(a) l'ensemble vide $\emptyset \in \mathcal{R}at(X^*)$ et $\forall x \in X, \{x\} \in \mathcal{R}at(X^*)$ (langages singletons)

(b) la famille est fermée pour les trois opérations dites **rationnelles** : l'union ($\cup$), le produit (.) et le passage à l'étoile ($*$).

1.2. Propriété

Nous avons vu que tout langage fini (et donc en particulier le langage vide et les langages singletons) est reconnaissable. Nous avons également montré que la famille des langages reconnaissables est fermée, en particulier, pour les opérations ratinnelles (union, produit et étoile).

En d'autres termes, la famille des langages rationnels est incluse dans la famille des langages reconnaissables ($\mathcal{R}at(X^*) \subset \mathcal{R}ec(X^*)$).

2. Les expressions rationnelles ou régulières

De la même manière qu'on définit des expressions arithmétiques ou booléennes, on définit des expressions qui permettent de repésenter les langages rationnels.

2.1. Syntaxe des expressions rationnelles

2.1.1. Règles de construction des expressions rationnelles

Nous donnons tout d'abord les règles syntaxiques de construction des **expressions rationnelles** ou **régulières** (désignées en abrégé par ER) sur un alphabet X :

- les expressions constantes sont $\emptyset$, ϵ et x ($\forall x \in X$) sont des ER ;
- des variables ayant comme valeur une expression rationnelle sont des ER (l'utilisation de telles variables permet d'alléger l'écriture des expressions) ;
- si E est une ER, (E) est une ER ;
- si E_1 et E_2 sont des ER, $(E_1 + E_2)$ est une ER ;
- si E_1 et E_2 sont des ER, $(E_1.E_2)$ est une ER (le point étant éventuellement omis) ;
- si E est une ER, E^* est une ER.

On construit ainsi des expressions telles que

$$(0+1)^* \quad ((0(10)^*)1) \quad ((10)^*(01)^*)$$

2.1.2. Conventions

Ainsi qu'on le fait dans le cas des expressions arithmétiques ou booléennes, des conventions permettent d'alléger l'écriture des expressions rationnelles en omettant le caractère . dans leur écriture et en y supprimant des parenthèses en supposant que (dans

2.2. Sémantique des expressions rationnelles

À chaque expression rationnelle on associe un langage au moyen de l'application μ suivante :

- $\mu[\emptyset] = \emptyset \qquad \mu[x] = \{x\} \qquad \mu[\epsilon] = \{\epsilon\}$
- $\mu[(E_1 + E_2)] = \mu[E_1] \cup \mu[E_2]$
- $\mu[(E_1.E_2)] = \mu[E_1].\mu[E_2]$
- $\mu[E^*] = (\mu[E])^*$

Ainsi, pour les expressions données en exemple, on a :

- $\mu[(0+1)^*] = \{0,1\}^*$
- $\mu[((0(10)^*)1)] = \{0\}.\{10\}^*.\{1\}$
- $\mu[((10)^*(01)^*)] = \{10\}^*.\{01\}^*$

2.3. Propriété fondamentale

Un langage L est rationnel si et seulement si il existe une expression rationnelle E telle que $L = \mu(E)$

On confondra souvent, par abus de langage, une expression et le langage qu'elle représente (omettant ainsi la fonction μ).

2.4. Langages et expressions

De même qu'un langage reconnaissable donné peut être reconnu par des automates différents (du point de vue de leurs structures), un même langage peut être représenté par différentes expressions (qu'on dit équivalentes).

Ainsi, les expressions $(a*b*)*$ et $(a+b)*$, bien que différentes, sont équivalentes et représentent toutes les deux le langage $\{a,b\}*$.

Pour montrer que deux expressions ne sont pas équivalentes, il suffit d'exhiber un mot appartenant au langage associé à l'une d'elle et n'appartenant pas au langage associé à l'autre.

Ainsi, les deux expressions $(aa+b)*$ et $(aa+aab+bb)*$ ne sont pas équivalentes car, par exemple, b appartient au langage associé à la première et pas à celui associé à la seconde.

Prouver que deux expressions sont équivalentes n'est pas toujours aussi simple que dans l'exemple d'expressions équivalentes donné précedemment. Nous reviendrons sur ce point ultérieurement.

3. Automates finis et expressions rationnelles

Nous allons montrer, par deux méthodes différentes, que tout langage reconnaissable est représentable par une expression rationnelle et donc que tout langage reconnaissable est rationnel (c'est-à-dire que $\mathcal{R}\mathrm{ec}(\mathrm{X}^*) \subset \mathcal{R}\mathrm{at}(\mathrm{X}^*)$).

Cela établira l'égalité des deux familles connue sous le nom de théorème de Kleene [$\mathcal{R}\mathrm{ec}(\mathrm{X}^*)$=$\mathcal{R}\mathrm{at}(\mathrm{X}^*)$)]

Nous reviendrons ultérieurement sur un procédé effectif permettant la construction d'un automate fini associé à une expression rationnelle donnée.

3.1. Automates finis et systèmes d'équations de langages

3.1.1. Notations

Étant donné un AFD $\mathcal{A} =< Q, X, \delta, q_0, Q' >$, nous notons $\forall q \in Q$,

$$L_q = \{w \mid w \in X^*, \delta^*(q, w) \in Q'\}$$

On a

$$L_q = \sum_{x \in X} x L_{\delta(q,x)}$$

et si $q \in Q'$ (le mot vide $\epsilon \in L_q$), on ajoute alors ϵ à la somme précédente.

Remarque : la définition des ensembles L_q est également possible en partant d'un AFND.

3.1.2. Le lemme d'Arden

Une équation de la forme $X = AX + B$ (dite linéaire) où A ne contient pas le mot vide ϵ a une solution unique qui est $A^*.B$

Preuve

- la vérification du fait que $A^*.B$ est une solution est immédiate ;
- pour montrer l'unicité de la solution, supposons qu'il en existe deux, X_1 et X_2.
  On a $X_1 - X_2 = (AX_1 \cup B) - (AX_2 \cup B) \subset AX_1 - AX_2 \subset A(X_1 - X_2)$.
  Supposons $X_1 - X_2 \neq \emptyset$: $\exists w \in X_1 - X_2$ de longueur minimale.
  Puisque $X_1 - X_2 \subset A(X_1 - X_2)$, $w \in A(X_1 - X_2)$ et donc $\exists u \in X_1 - X_2$ et $\exists a \in A$ tels que $w = a.u$
  Puisque $\epsilon \notin A$, on a $\mid u \mid < \mid w \mid$, ce qui contredit l'hypothèse de minimalité de longueur faite sur w. Donc $X_1 - X_2 = \emptyset$. On démontre de même que $X_2 - X_1 = \emptyset$.
  D'où $X_1 = X_2$ et l'unicité de la solution.

3.1.3. Application : résolution d'un système d'équations linéaires :

Un système de n équations linéaires gauches, à n inconnues X_i, de la forme

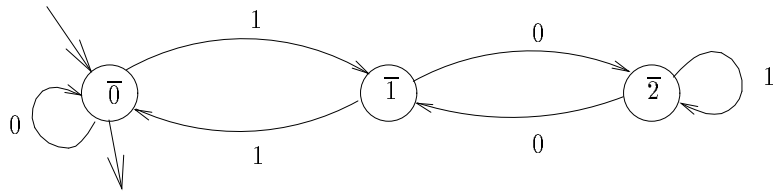
$$X_i = \sum_{j=1}^{j=n} A_{i,j}.X_j + B_i$$

tel que celui obtenu à partir d'un AFD peut être résolu par élimination successive des variables en s'appuyant sur le lemme d'Arden.

Si les $A_{i,j}$ et les B_i sont des langages rationnels (c'est le cas pour les systèmes construits à partir d'automates), les solutions obtenues pour les variables X_i le sont aussi.

3.1.4. Exemple : les multiples de 3 écrits en base 2

Nous avons vu qu'un automate reconnaissant le langage formé de tous les mots sur l'alphabet $\{0, 1\}$ correspondant à la représentation binaire d'un multiple de 3 était :



On peut lui associer le système de trois équations à 3 inconnues L_0, L_1 et L_2 :

$$L_0 = 0.L_0 + 1.L_1 + \epsilon$$
$$L_1 = 0.L_2 + 1.L_0$$
$$L_2 = 0.L_1 + 1.L_2$$

où la variable L_0 correspond au langage reconnu par l'automate.

- **Première résolution possible**

  L'application du lemme d'Arden à la dernière équation donne

  $$L_2 = 1^*0L_1$$

  En reportant cette valeur dans la seconde équation, on obtient

  $$L_1 = 01^*0L_1 + 1.L_0$$

  L'application du lemme d'Arden à cette équation donne :

  $$L_1 = (01^*0)^*1L_0$$

  En reportant cette valeur de L_1 dans la première équation on obtient :

  $$L_0 = 0.L_0 + 1(01^*0)^*1L_0 + \epsilon = (0 + 1(01^*0)^*1)L_0 + \epsilon$$

  ce qui donne en appliquant à nouveau le lemme d'Arden

  $$L_0 = (0 + 1(01^*0)^*1)^*\epsilon = (0 + 1(01^*0)^*1)^* \ (\epsilon \text{ élément neutre de } .)$$

- **Une autre résolution possible**

  On commence par éliminer L_1, ce qui donne :

  $$L_0 = 0L_0 + 1(0L_2 + 1L_0) + \epsilon = (0 + 11)L_0 + 10L_2 + \epsilon$$
  $$L_2 = 0(0L_2 + 1L_0) + 1L_2 = (00 + 1)L_2 + 1L_0$$

  Par application du lemme d'Arden on obtient $L_2 = (00 + 1)^*1L_0$, puis par report dans la première équation et enfin application du lemme d'Arden :

  $$\begin{aligned} L_0 &= (0 + 11)L_0 + 10(00 + 1)^*1L_0 + \epsilon \\ &= (0 + 11 + 10(00 + 1)^*1)L_0 + \epsilon \\ &= (0 + 11 + 10(00 + 1)^*1)^*\epsilon \\ &= (0 + 11 + 10(00 + 1)^*1)^* \ (\epsilon \text{ élément n eutre de } .) \end{aligned}$$

On a ainsi obtenu deux expressions rationnelles :

$\hookrightarrow (0 + 1(01^*0)^*1)^*$

$\hookrightarrow (0 + 11 + 10(00 + 1)^*1)^*$

qui représentent le même langage et sont donc équivalentes.

3.2. L'algorithme de Mc-Naughton et Yamada

Nous décrivons maintenant une méthode différente de calcul d'une expression rationnelle décrivant le langage reconnu par un automate fini déterministe qui s'apparente au calcul de la fermeture transitive d'un graphe valué (c'est-à-dire au calcul des valuations de tous les chemins possibles dans ce graphe).

Le principe en est le suivant :

- ↪ soit $\mathcal{A} = < Q, X, \delta, q_0, Q' >$, avec $|Q| = n$. On suppose l'automate complet ;
- ↪ on numérote (arbitrairement) les n états de l'AFD avec les entiers 1, 2, ... , n et dans la suite on utilise ces numéros pour désigner les états (de fait on a simplement procédé à un renommage des états de l'automate sans en changer la structure) ;
- ↪ $\forall i, j \in [1 : n]$, $\forall k \in [0 : n]$, on désigne par $L_{i,j}^k$ le sous-ensemble de X^* défini par :

  $\{m \ / \ \delta^*(i,m)=j$ et $\forall u \in X^* - \{\epsilon, m\}$ facteur gauche de m, $\delta^*(i,u) \leq k\}$. Du point de vue du graphe associé à l'automate, m décrit donc un chemin de i à j qui ne passe par aucun sommet supérieur à k, mis à part éventuellement au début ou à la fin.
- ↪ i_0 étant l'état initial, on a $L(\mathcal{A}) = \bigcup_{j \in Q'} L_{i_0,j}^n$
- ↪ les ensembles $L_{i,j}^k$ se calculent par récurrence sur k :
  - $L_{i,j}^0 = \{x \ / \ x \in X$ et $\delta(i,x) = j\}$ $\quad (+ \{\epsilon\}$ si $i = j)$
  - $L_{i,j}^{k+1} = L$