Licence ST Informatique – S4 – 2008/2009

Automates et langages

mai 2009

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

Le chercheur russe Victor Glushkov a défini en 1961 un algorithme de construction d'un automate à partir d'une expression rationnelle.

1 Principe de la construction

- On associe à chaque lettre de l'expression rationnelle e une position.
- On définit un état pour chaque position i de l'expression, et on ajoute un état initial 0.
- On place une transition de i vers j étiquetée par α_j si et seulement si la lettre α_j peut suivre la lettre α_i dans un mot du langage $L(e)$.
- On place une transition de 0 vers i si et seulement si la lettre α_i peut être la première lettre d'un mot du langage $L(e)$.
- L'état i ($i > 0$) est final si et seulement si α_i peut être la dernière lettre d'un mot du langage $L(e)$.
- L'état 0 est final si et seulement si ε est dans le langage.

Exemple 1 *$(ab+c)^*ab$ est traduite en $(a_1b_2+c_3)^*a_4b_5$. L'automate de Glushkov de cette expression possède 6 états. L'état 0 est initial, l'état 5 est final. Sa table de transition est la suivante :*

	a	b	c
0	$\{1,4\}$	$\emptyset$	$\{3\}$
1	$\emptyset$	$\{2\}$	$\emptyset$
2	$\{1,4\}$	$\emptyset$	$\{3\}$
3	$\{1,4\}$	$\emptyset$	$\{3\}$
4	$\emptyset$	$\{5\}$	$\emptyset$
5	$\emptyset$	$\emptyset$	$\emptyset$

2 Construction de l'automate

Pour construire l'automate de Glushkov associé à une expression rationnelle E, on va définir plusieurs fonctions relatives à la notion de position dans E.

1. `first`(E) est l'ensemble des positions dont les lettres peuvent commencer un mot de $L(E)$
2. `last`(E) est l'ensemble des positions dont les lettres peuvent terminer un mot de $L(E)$
3. `nullable`(E) est un prédicat indiquant si $L(E)$ contient le mot vide.
4. `follow`(E, x) où x est une position est l'ensemble des positions qui peuvent suivre la position x dans un mot de E. Cette fonction définit la fonction de transition de l'automate de Glushkov.

L'expression rationnelle peut être vue comme un arbre, où les feuilles sont étiquetées par $\emptyset$, ε, ou les positions des symboles, et les noeuds internes sont étiquetés par les opérateurs union ($+$), étoile (*), et concaténation ($.$). A chaque noeud de l'arbre, on associe quatre attributs :

- `nullable` de type boolean
- `first` et `last` qui sont des ensembles de positions
- `follow` qui est un ensemble de couples (position, ensemble de positions)

et voici comment les calculer en fonction du noeud ν considéré :

- si ν est un noeud $\emptyset$

```
nullable = false ,
first, last et follow sont des ensembles vides.
```

- si ν est un noeud ε

```
nullable = true
first, last et follow sont des ensembles vides
```

- si ν est un noeud x

```
nullable = false ,
first et last vallent {x},
follow(x) est l'ensemble vide.
```

- si ν est un noeud $+$

```
nullable = expGauche.nullable OR expDroite.nullable,
first = expGauche.first UNION expDroite.first,
last = expGauche.last UNION expDroite.last,
follow(x) = expGauche.follow(x) si x position de expGauche
            = expDroite.follow(x) si x position de expDroite
```

- si ν est un noeud $.$

```
nullable = expGauche.nullable AND expDroite.nullable,

si expGauche.nullable alors first = expGauche.first UNION expDroite.first,
                          sinon first = expGauche.first
si  expDroite.nullable alors last = expGauche.last UNION expDroite.last,
                          sinon last = expDroite.last,
pour chaque x dans expGauche.last faire
                  follow(x) = expGauche.follow(x) UNION expDroite.first
pour chaque x qui n'est pas dans expGauche.last faire
                  follow(x) = expGauche.follow(x) si x position de expGauche
                              = expDroite.follow(x) si x position de expDroite
```

- si ν est un noeud $*$

```
nullable = true
first = expFils.first
last = expFils.last
pour chaque x dans expFils.last faire
               follow(x) = expFils.follow(x) UNION expFils.first
pour chaque x qui n'est pas dans expFils.last faire
               follow(x) = expFils.follow(x)
```

Vous trouverez une synthèse des différents algorithmes de construction des automates de Glushkov, et l'étude de leur complexité dans :

D. Ziadi, J.-L. Ponty et J.-M. Champarnaud, Passage d'une expression rationnelle à un automate fini non-déterministe, Bulletin of the Belgian Mathematical Society Simon Stevin, 4-2(1997), pages 177–203