TD5 - Automates et Langages

octobre 2018

First, Last, Follow – Automate de Glushkov

Exercice 1 :
Pour tout langage L sur un alphabet Σ, on définit les ensembles suivants

$$Eps(L) = L \cap \{\varepsilon\}$$

$$First(L) = \{x \in \Sigma, \exists u \in \Sigma^*, xu \in L\} = \{x \in \Sigma, \{x\}.\Sigma^* \cap L \neq \emptyset\}$$
$$Last(L) = \{x \in \Sigma, \exists u \in \Sigma^*, ux \in L\} = \{x \in \Sigma, \Sigma^*.\{x\} \cap L \neq \emptyset\}$$
$$Follow(L) = \{f \in \Sigma^2, \exists u \in \Sigma^*, \exists v \in \Sigma^*, ufv \in L\} = \{f \in \Sigma^2, \Sigma^*.\{f\}.\Sigma^* \cap L \neq \emptyset\}$$

En d'autres termes $First(L)$ est l'ensemble des préfixes de longueur 1, $Last(L)$ l'ensemble des suffixes de longueur 1, et $Follow(L)$ l'ensemble des facteurs de longueur 2 de mots de L.
Q 1 . Indiquez les valeurs de $First$, $Last$, $Follow$, Eps pour chacun des langages dénotés par les expressions rationnelles : $abc + ab + abcab$, $a(a+c)^*b$, ac^*b, ϵ, $\emptyset$, a
Q 2 . Soient deux langages L_1 et L_2. Exprimez $First(L_1 \cup L_2)$, $First(L_1.L_2)$, $First(L_1^*)$ en fonction de $First(L_1)$, $First(L_2)$ et $Eps(L_1)$
Q 3 . Faites de même pour $Last$, puis pour $Follow$
Q 4 . Soit e une expression rationnelle de longueur > 1. On veut calculer les ensembles $First$, $Last$ et $Follow$ du langage dénoté par e. Montrez que l'on peut toujours se ramener au calcul de ces mêmes ensembles pour des expressions rationnelles strictement plus courtes que e.
Q 5 . En utilisant les résultats de la question 2, écrivez un algorithme récursif calculant les 4 ensembles correspondant à une expression rationnelle quelconque. Il prendra la forme d'une fonction récursive $FLFE(e)$, dont le paramètre e est une expression et dont le résultat est un quadruplet $(First, Last, Follow, Eps)$.

Exercice 2 :
Une expression rationnelle est dite linéaire si aucune lettre n'y apparaît plus d'une fois. Par exemple $a(b+c)^*$, $(ca)^*$, $a+c$ sont linéaires mais $a(b+c)^*a$ ou $(ca)^*+c$ ne le sont pas.
À toute expression rationnelle e définie sur un alphabet Σ, on peut associer une expression linéarisée e', construite sur un alphabet Σ' élargi, de la façon suivante :

- on numérote à partir de 1 toutes les positions de lettres dans l'expression .
- on remplace chaque occurrence d'une lettre x par x_i où i est le numéro de la position qu'elle occupe. Cette expression utilise l'alphabet $\{x_i, x \in \Sigma, x$ est présente à la position i dans l'expression$\}$

Par exemple $a(b+c)^*a$ se linéarisera en $a_1(b_2+c_3)^*a_4$ et $(ca)^*+c$ en $(c_1a_2)^*+c_3$
Q 1 . Linéariser l'expression $e = (ab)^*ab(ca+b)^*$
Q 2 . Calculer les ensembles First, Last, et Follow de l'expression linéarisée.
Q 3 . Pour toute expression linéarisée e où interviennent n lettres, on définit un automate déterministe à $n+1$ états ($\mathcal{Q} = [0, n]$) dont 0 est l'état initial. Les transitions sont définies par

- $\delta(0, x_i) = i, \forall x_i \in First(e)$
- $\delta(j, x_i) = i, \forall j > 0, \forall x_j x_i \in Follow(e)$

Enfin, les états acceptants seront $\mathcal{F} = \{i, x_i \in Last(e)\} \cup Eps(e).\{0\}$

Construire l'automate obtenu pour l'expression définie à la question 1.
Q 4 . À partir de cet automate déterministe, on obtient un automate (potentiellement non déterministe) sur l'alphabet Σ en supprimant tous les indices des lettres.
Dessinez cet automate.
Il reconnaît le langage dénoté par l'expression initiale e

La construction que nous venons de réaliser est appelée la « **construction de Glushkov** » et l'automate obtenu « **automate de Glushkov** ».
Cette construction qui peut s'appliquer à toute expression rationnelle e constitue un algorithme de calcul d'un automate reconnaissant le langage dénoté par e.