Université Paris Cité
Automates et Analyse Lexicale

UFR Informatique
Année 2023–2024

# Handout 6

L'algorithme de Glushkov

## 1 Algorithme de Glushkov

Cet algorithme transforme une expression rationnelle directement en un automate non-déterministe, sans de produire des $\epsilon$-transitions.

### 1.1 Linéariser l'expression rationnelle

Une expression rationnelle $r$ sur l'alphabet $\Sigma$ est *linéaire* quand chaque symbole de $\Sigma$ paraît au plus une fois dans $r$. Si une expression rationnelle n'est pas linéaire on peut la linéariser en passant à un alphabet *plus grand*, appelé l'*alphabet linéarisé*.

La méthode que nous proposons içi est : on remplaçe les occurrences différentes de lettres par des nombres consécutifs à partir de 1. Si $r$ contient $n$ occurrences de lettres, l'expression linéarisée $r'$ sera donc sur l'alphabet $\Sigma' = \{1, \ldots, n\}$. On notera dans la suite $l(i)$ $(1 \leq i \leq n)$ pour la $i$-ème lettre de $r$.

Exemple : $r = (ab + b)^*(bb + a^*)$ sur l'alphabet $\Sigma = \{a, b\}$ devient $r' = (12 + 3)^*(45 + 6^*)$ sur l'alphabet $\Sigma' = \{1, 2, 3, 4, 5, 6\}$, et on a $l(1) = a$, $l(2) = b$, $l(3) = b$, etc.

### 1.2 Analyser l'expression rationnelle linéarisée

Calculer les informations suivantes sur l'expression rationnelle linéarisée $r'$ :

- est-ce que $\epsilon \in \mathcal{L}(r')$ ?
- $first(r') = \{a \in \Sigma' \mid \exists w \in \Sigma'^* : aw \in \mathcal{L}(r')\}$
- $last(r') = \{a \in \Sigma' \mid \exists w \in \Sigma'^* : wa \in \mathcal{L}(r')\}$
- $next(r') = \{f \in \Sigma'^2 \mid \exists v, w \in \Sigma'^* : v \cdot f \cdot w \in \mathcal{L}(r')\}$

Souvent on se contente de dire qu'on peut "voir" ces informations. Des définitions formelles sont données à la section 1.5.

### 1.3 Construire un automate non déterministe

L'automate est défini comme suit :

- l'alphabet est $\Sigma$, donc l'alphabet *avant* la linéarisation.
- ensemble d'états $Q = \{0, 1, \ldots, n\}$, où $n$ est le nombre de lettres dans l'alphabet linéarisé.
- état initial $q_0 = 0$
- états acceptants : $F = last(r')$, plus l'état 0 dans le cas où $\epsilon \in \mathcal{L}(r')$

- il y a deux types de transitions :
  - quand $i \in first(r')$, alors transition de 0 vers $i$ par la lettre $l(i)$ ;
  - quand $ij \in next(r')$, alors transition de $i$ vers $j$ par la lettre $l(j)$.

### 1.4 Déterminiser

Comme vu au cours 3.

### 1.5 Définition formelle de l'analyse de l'expression rationnelle

- La fonction *eps*: $\textsc{ExpRat} \to \{\texttt{true}, \texttt{false}\}$

$$\begin{array}{rcll} eps(\epsilon) & = & \texttt{true} & \\ eps(a) & = & \texttt{false} & a \in \Sigma \\ eps(\emptyset) & = & \texttt{false} & \\ eps(r_1 + r_2) & = & eps(r_1) \vee eps(r_2) & \\ eps(r_1 r_2) & = & eps(r_1) \wedge eps(r_2) & \\ eps(r^*) & = & \texttt{true} & \end{array}$$

- La fonction *first*: $\textsc{ExpRat} \to \mathcal{P}(\Sigma)$

$$\begin{array}{rcll} first(\epsilon) & = & \emptyset & \\ first(a) & = & \{a\} & a \in \Sigma \\ first(\emptyset) & = & \emptyset & \\ first(r_1 + r_2) & = & first(r_1) \cup first(r_2) & \\ first(r_1 r_2) & = & \begin{cases} first(r_1) & \text{si } \neg eps(r_1) \\ first(r_1) \cup first(r_2) & \text{si } eps(r_1) \end{cases} & \\ first(r^*) & = & first(r) & \end{array}$$

- La fonction *last*: $\textsc{ExpRat} \to \mathcal{P}(\Sigma)$

$$\begin{array}{rcll} last(\epsilon) & = & \emptyset & \\ last(a) & = & \{a\} & a \in \Sigma \\ last(\emptyset) & = & \emptyset & \\ last(r_1 + r_2) & = & last(r_1) \cup last(r_2) & \\ last(r_1 r_2) & = & \begin{cases} last(r_2) & \text{si } \neg eps(r_2) \\ last(r_2) \cup last(r_1) & \text{si } eps(r_2) \end{cases} & \\ last(r^*) & = & last(r) & \end{array}$$

- La fonction *next*: $\textsc{ExpRat} \to \mathcal{P}(\Sigma^2)$

$$\begin{array}{rcll} next(\epsilon) & = & \emptyset & \\ next(a) & = & \emptyset & a \in \Sigma \\ next(\emptyset) & = & \emptyset & \\ next(r_1 + r_2) & = & next(r_1) \cup next(r_2) & \\ next(r_1 r_2) & = & next(r_1) \cup next(r_2) \cup last(r_1)first(r_2) & \\ next(r^*) & = & next(r) \cup last(r)first(r) & \end{array}$$