AUTOMATE FINITE ȘI LIMBAJE REGULATE

Pentru gramaticile de tipul 3, sau regulate, am convenit asupra formei regulilor (la Clasificarea Chomsky).

$$\begin{cases} A \to pB \\ C \to q \end{cases}, \quad p, q \in V_T^*, \quad A, B, C \in V_N$$

<u>DEFINIȚIE</u> Spunem că o gramatică regulată este în <u>formă normală</u> dacă regulile au forma

$$\begin{cases} A \to iB \\ C \to j \end{cases}, \quad A, B, C \in V_N, \quad i, j \in V_T$$

Se admite și $S \to \lambda$, dar în acest caz S nu apare în dreapta nici unei alte reguli.

<u>LEMĂ</u>: $\forall G \in \mathcal{G}_3 \ \exists G' \in \mathcal{G}_3$ în formă normală a.î. $L(G) = L(G')$

(orice gramatică regulată admite o formă normală)

\#

Fie $G = (V_N, V_T, x_0, \mathcal{P}) \in \mathcal{G}_3$ și $L = L(G)$.

Construim o gramatică echivalentă în formă normală urmând pașii:

1) Eliminăm regulile de ștergere (cu algoritmul prezentat la incluziunea $\mathcal{L}_2 \subset \mathcal{L}_1$)

2) Eliminăm "redenumirile" (reguli de forma $A \to B$) astfel

Dacă $A \to B \in \mathcal{P}$ și $B \to \alpha_1 | \dots | \alpha_k \in \mathcal{P}$ atunci redenumirea se înlocuiește cu $A \to \alpha_1 | \dots | \alpha_k \in \mathcal{P}'$

(evident regulile lui B se păstrează)

3) Scurtăm regulile neconvenabile

Dacă $A \to i_1 i_2 \dots i_k B \in \mathcal{P}$ atunci folosim șirul de reguli noi în $\mathcal{P}'$

$$\begin{cases} A \to i_1 Z_1 \\ Z_1 \to i_2 Z_2 \\ \vdots \\ Z_{k-1} \to i_k B \end{cases} \in \mathcal{P}, \text{ cu } Z_1, Z_2, \dots, Z_{k-1} \in V_N' \text{ neterminale noi}$$

Idem $C \to j_1 j_2 \dots j_k \in \mathcal{P}$ înlocuit cu $\begin{cases} C \to j_1 Z_1 \\ \vdots \\ Z_{k-1} \to j_k \end{cases} \in \mathcal{P}'$

Exemplu: Fie limbajul $L = \{abaw \mid w \in \{a,b\}^*\} \cup \{\lambda\} \in \mathcal{L}_3$ generat de gramatica cu regulile

$$G: \begin{cases} A \to abaB \mid \lambda \\ B \to aB \mid bB \mid \lambda \end{cases}$$

Construim gramatica echivalentă în formă normală

1. Elimin regulile de ștergere

$A, B \in U_f$ (pot genera λ).

Gram. modificata $G_1 \begin{cases} A \to abaB \mid aba \\ B \to aB \mid a \mid bB \mid b \end{cases}$ $L(G_1) \not\ni \lambda$

Adaug un nou neterminal pt producerea lui λ.

$$G_2: \begin{cases} S \to \lambda \mid A \\ A \to abaB \mid aba \\ B \to aB \mid bB \mid a \mid b \end{cases} \quad L(G_2) = L = L(G)$$

2. Elimin redenumiri — În cazul nostru "$S \to A$"!

$$G_3: \begin{cases} S \to \lambda \mid abaB \mid aba \\ A \to abaB \mid aba \\ B \to aB \mid bB \mid a \mid b \end{cases}$$

3. Scurtez reguli neconvenabile

$S \to abaB$ se înloc. cu $\begin{cases} S \to aZ_1 \\ Z_1 \to bZ_2 \\ Z_2 \to aB \end{cases}$

$S \to aba$ se înloc. cu $\begin{cases} S \to aT_1 \\ T_1 \to bT_2 \\ T_2 \to a \end{cases}$

$A \to abaB$ se înloc cu $\begin{cases} A \to a\Theta_1 \\ \Theta_1 \to b\Theta_2 \\ \Theta_2 \to aB \end{cases}$

$A \to aba$ se înloc. cu $\begin{cases} A \to a\ddot{U}_1 \\ \ddot{U}_1 \to b\ddot{U}_2 \\ \ddot{U}_2 \to a \end{cases}$

Restul regulilor se păstrează

Obs: Putem refolosi unele neterminale noi. De ex Z_1 în loc de Θ_1
$Z_2 \Leftrightarrow \Theta_2$ $T_1 \Leftrightarrow \ddot{U}_1$ $T_2 \Leftrightarrow \ddot{U}_2$

TEOREMA $\mathcal{R} = \mathcal{L}_3$ ($\mathcal{R}$ - fam. lb. rec de AF)

#

Partea I $\mathcal{L}_3 \subseteq \mathcal{R}$

Fie $L \in \mathcal{L}_3$ și $G = (V_N, V_T, x_0, P) \in \mathcal{G}_3$ în formă normală a.î. $L = L(G)$.

Regulile lui G au forma $\begin{cases} A \to iB \\ C \to j \end{cases}$, $i, j \in V_T$.

(Cazul $\lambda \in L$ îl tratăm separat)

Construim următorul $AF = (S, I, f, s_0, S_f)$ unde

$S = V_N \cup \{X\}$, X - un simbol nou introdus $X \notin V_N$

$I = V_T$

$s_0 = x_0$

$S_f = \{X\}$

Dacă $A \to iB \in P$ atunci $f(A, i) \ni B$

(pe diagrama AF desenăm un arc cu eticheta i de la nodul A la nodul B $A \xrightarrow{i} B$)

Dacă $C \to j \in P$ atunci $f(C, j) \ni X$

(pe diagramă conectez nodul C cu starea finală X.)

Rămâne de arătat că $L(AF) = L(G)$.

#

Fie $p = i_1 i_2 \ldots i_k \in L(G)$ $k \geq 1$. ptr că $\lambda \notin L$

Detaliem derivarea ce produce p.

(1) $x_0 \Rightarrow i_1 X_1 \Rightarrow i_1 i_2 X_2 \Rightarrow \ldots \Rightarrow i_1 i_2 \ldots i_{k-1} X_{k-1} \Rightarrow i_1 i_2 \ldots i_k$

Șirul de reguli aplicate este

(2) $\begin{cases} x_0 \to i_1 X_1; \\ X_1 \to i_2 X_2; \in P \\ \vdots \\ X_{k-1} \to i_k; \end{cases}$

În AF avem arcele (3) $\begin{cases} f(x_0, i_1) \ni X_1 \\ f(X_1, i_2) \ni X_2 \\ \vdots \\ f(X_{k-1}, i_k) \ni X \end{cases}$

Cu arcele (3) putem construi traiectoria AF

$$(4) \quad x_0 \xrightarrow{i_1} X_1 \xrightarrow{i_2} X_2 \longrightarrow \dots \longrightarrow X_{k-1} \xrightarrow{i_k} X \in S_f.$$

Deci $p \in L(AF)$

Invers, dacă $p = i_1 \dots i_k \in L(AF)$ avem o traiectorie de recunoaștere (4), deci avem șirul de arce (3) cărora le corespund regulile (2). Putem construi derivarea (1) deci $p \in L(G)$. q.e.d.

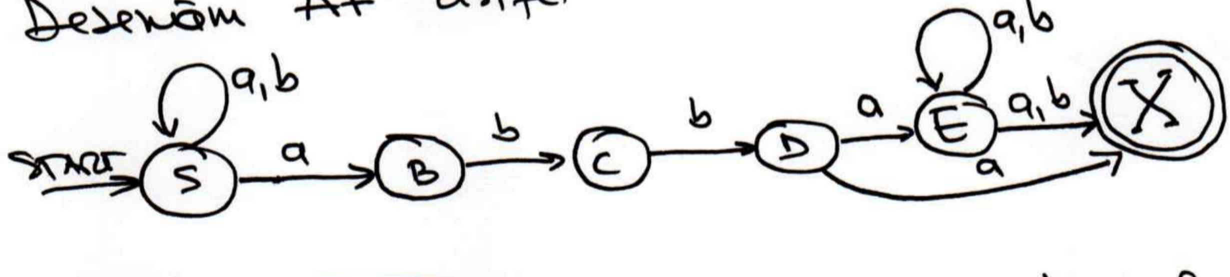
<u>Exemplu</u>: Gramatica G cu regulile

$$\begin{cases} S \to aS \mid bS \mid aB \\ B \to bC \\ C \to bD \\ D \to aE \mid a \\ E \to aE \mid bE \mid a \mid b \end{cases}$$

generează limbajul

$$L = \{ w\, abba\, w' \mid w, w' \in \{a, b\}^* \}$$

Desenăm AF astfel



$\boxed{\text{Cazul } \lambda \in L}$

Pt. un AF recunoașterea lui λ înseamnă $f(s_0, \lambda) \in S_f$, adică $s_0 \in S_f$.

Dacă $\lambda \in L$ atunci $S_f = \{X, x_0\}$

<u>Obs</u>: Corespondența între generare și recunoaștere pentru un cuvânt este între <u>derivarea</u> ce produce cuvântul și <u>traiectoria</u> de recunoaștere

Partea II $\mathcal{R} \subseteq \mathcal{L}_3$

Fie $L \in \mathcal{R}$ şi $AFD = (S, I, f, s_0, S_f)$ aî $L = L(AFD)$

Construim gramatica $G = (V_N, V_T, x_0, \mathcal{P})$ astfel

$V_N = S$

$V_T = I$

$x_0 = s_0$

Dacă $f(s, i) = s'$ atunci $s \to is' \in \mathcal{P}$

(Dacă am arc de la s la s' $\quad s \xrightarrow{i} s'$
atunci $s \to is' \in \mathcal{P}$ e regulă)

Dacă $f(s,i) = s'$ şi $s' \in S_f$ atunci
adaug două reguli $s \to is' \in \mathcal{P}$ şi $s \to i$

(Dacă $\quad s \xrightarrow{i} s' \quad$ at. $\left\{ \begin{array}{l} s \to is' \\ s \to i \end{array} \right. \in \mathcal{P}$)

Demonstraţia $L(AFD) = L(G)$ este similară celei de la partea I, deci rămâne ca exerciţiu pentru doritori.

<u>Exemplu</u> Limbajul $L = \{ w \in \{0,1\}^* \mid w \text{ conţine cel puţin un } 0 \}$

este recunoscut de următorul AFD

Regulile gramaticii echivalente sunt $\left\{ \begin{array}{l} A \to 1A \mid 0B \mid 0 \\ B \to 0B \mid 0 \mid 1B \mid 1 \end{array} \right.$

<u>Obs</u>: 1) Construcţia din demonstraţie rămâne valabilă dacă se porneşte cu un AF nedeterminist

2) Dacă $s_0 \in S_f$ atunci adăugăm regula $s_0 \to \lambda \in \mathcal{P}$ pentru a produce şi cuvântul λ!