TEHNICI DE COMPILARE – CURSUL 7

EXEMPLU DE GRAMATICĂ $SLR(1)$

Fie G_2 gramatica cu producțiile:

1: $E \rightarrow TR$
2: $R \rightarrow +TR$
3: $R \rightarrow *TR$
4: $R \rightarrow \lambda$
5: $T \rightarrow n$
6: $T \rightarrow (E)$

1) Extindem G la G': introducem productia $E' \rightarrow E$
2) Calculăm mulțimile $Follow$ cu algoritmul din cursul 4, inițializând $Follow(E)$ cu $\{\#\}$. Obținem:

$Follow(X)$	Pasul 1	Pasul 2
E	$\#,)$	
T	$+, *, \#$	$)$
R	$\#$	$)$

3) Calculam mulțimile canonice $LR(0)$ pentru G_2:

$$I_0 = \begin{bmatrix} E' \rightarrow .E \xrightarrow{goto(I_0,E)} I_1 \\ E \rightarrow .TR \xrightarrow{goto(I_0,T)} I_2 \\ T \rightarrow .n \xrightarrow{goto(I_0,n)} I_3 \\ T \rightarrow .(E) \xrightarrow{goto(I_0,()} I_4 \end{bmatrix}$$

$$I_2 = \begin{bmatrix} E \rightarrow T.R \xrightarrow{goto(I_2,R)} I_5 \\ R \rightarrow .+TR \xrightarrow{goto(I_2,+)} I_6 \\ R \rightarrow .*TR \xrightarrow{goto(I_2,*)} I_7 \\ R \rightarrow . \end{bmatrix}$$

$$I_1 = [E' \rightarrow E.]$$

$$I_3 = [T \rightarrow n.]$$

$$I_4 = \begin{bmatrix} T \to (.E) \xrightarrow{goto(I_4,E)} I_8 \\ E \to .TR \xrightarrow{goto(I_4,T)} I_2 \\ T \to .n \xrightarrow{goto(I_4,n)} I_3 \\ T \to .(E) \xrightarrow{goto(I_4,()} I_4 \end{bmatrix}$$

$$I_5 = [E \to TR.]$$

$$I_6 = \begin{bmatrix} R \to +.TR \xrightarrow{goto(I_6,T)} I_9 \\ T \to .n \xrightarrow{goto(I_6,n)} I_3 \\ T \to .(E) \xrightarrow{goto(I_6,()} I_4 \end{bmatrix}$$

$$I_7 = \begin{bmatrix} R \to *.TR \xrightarrow{goto(I_7,T)} I_{10} \\ T \to .n \xrightarrow{goto(I_7,n)} I_3 \\ T \to .(E) \xrightarrow{goto(I_7,()} I_4 \end{bmatrix}$$

$$I_8 = \left[T \to (E.) \xrightarrow{goto(I_8,))} I_{11} \right]$$

$$I_9 = \begin{bmatrix} R \to +T.R \xrightarrow{goto(I_9,R)} I_{12} \\ R \to .+TR \xrightarrow{goto(I_9,+)} I_6 \\ R \to .*TR \xrightarrow{goto(I_9,*)} I_7 \\ R \to . \end{bmatrix}$$

$$I_{10} = \begin{bmatrix} R \to *T.R \xrightarrow{goto(I_{10},R)} I_{13} \\ R \to .+TR \xrightarrow{goto(I_{10},+)} I_6 \\ R \to .*TR \xrightarrow{goto(I_{10},*)} I_7 \\ R \to . \end{bmatrix}$$

$$I_{11} = [T \to (E).]$$

$$I_{12} = [R \to +TR.]$$

$$I_{13} = [R \to *TR.]$$

Tabela de analiză sintactică $SLR(1)$ pentru G_2:

	action						*goto*		
	+	*	n	(	)	#	E	T	R
0	Error	Error	shift 3	shift 4	error	error	1	2	Error
1	Error	Error	error	error	error	accept	error	error	Error
2	shift 6	shift 7	error	error	reduce 4	reduce 4	error	error	5
3	reduce 5	reduce 5	error	error	reduce 5	reduce 5	error	error	Error
4	Error	Error	shift 3	shift 4	error	error	8	2	Error
5	Error	Error	error	error	reduce 1	reduce 1	error	error	Error
6	Error	Error	shift 3	shift 4	error	error	error	9	Error
7	Error	Error	shift 3	shift 4	error	error	error	10	Error
8	Error	Error	error	error	shift 11	error	error	error	Error
9	shift 6	shift 7	error	error	reduce 4	reduce 4	error	12	error
10	shift 6	shift 7	error	error	reduce 4	reduce 4	error	13	error
11	reduce 6	reduce 6	error	error	reduce 6	reduce 6	error	error	error
12	Error	Error	error	error	reduce 2	reduce 2	error	error	error
13	Error	Error	error	error	reduce 3	reduce 3	error	error	error

Tabela $action$ nu are intrări multiple, rezultă că G_2 este $SLR(1)$.

Analiza sintactică decurge exact ca pentru gramaticile $LR(1)$.

PROPRIETĂȚI ALE GRAMATICILOR DE TIP *LR*

Teorema 1. Automatul finit determinist construit pe baza funcției $goto$ din algoritmul $LR(1)$ recunoaște mulțimea prefixelor viabile ale lui G.

Demonstrație. Fie $G = (N, \Sigma, S, P)$ gramatică independentă de context și extensia sa definita prin $G' = (N \cup \{S'\}, \Sigma \cup \{\#\}, S', P \cup \{S' \to S\})$ cu $L(G) = L(G')$. Considerăm automatul $A = (Q, \Sigma \cup N, goto, I_0, F)$, unde $Q = C_G = \{I_0, I_1, \dots, I_n\}$, $I_0 = closure(\{S' \to .S; \#\}), goto$ fiind funcția de tranziție a lui A, C_G mulțimile canonice $LR(1)$ asociate lui $G, F = \{J \in C_G | \exists A \to \alpha.; a \in J\}$.

Arătăm că $\forall A \neq S', A \to \alpha.\beta; a \in goto(I_0, \gamma) \Leftrightarrow A \to \alpha.\beta; a$ validă pentru prefixul viabil $\gamma \in (N \cup \Sigma)^*$.

"$\Longrightarrow$" $|\gamma| = n$. Facem inducție după n.

$n = 0$. Atunci $\gamma = \lambda$ și $goto(I_0, \gamma) = I_0 = closure(\{S' \to .S; \#\})$, deci $\alpha = \lambda$ și $A \to \alpha.\beta; a \in I_0 = closure(\{S' \to .S; \#\})$. Atunci există

$$
\begin{aligned}
&S \to \alpha_1, && \alpha_1 = A_1\beta_1 \\
&A_1 \to \alpha_2, && \alpha_2 = A_2\beta_2 \\
&\quad\vdots \\
&A_{k-1} \to \alpha_k, && \alpha_k = A_k\beta_k \\
&A_k \to \beta,
\end{aligned}
$$

unde $A \to \alpha.\beta; a = A_k \to \beta; a \in I_0$ (posibil $k = 0, \beta = \alpha_1$). Atunci:

$S \Rightarrow \alpha_1 = A_1\beta_1 \Rightarrow A_2\beta_2\beta_1 \Rightarrow \cdots \Rightarrow A_k\beta_k \dots \beta_1 \Rightarrow \beta\beta_k \dots \beta_1$ și $a \in First(\beta_k \dots \beta_1 \#)$. Fie $\beta_k \dots \beta_1 \underset{d}{\overset{*}{\Rightarrow}} \beta'_k \dots \beta'_1$, unde $\beta_i \underset{d}{\overset{*}{\Rightarrow}} \beta'_i, \beta'_i \in \Sigma^*$, $a = First(\beta'_k \dots \beta'_1 \#)$. Avem:

$$S \underset{d}{\overset{*}{\Rightarrow}} A\beta'_k \dots \beta'_1 \underset{d}{\Rightarrow} \beta\beta'_k \dots \beta'_1$$

$\gamma = \lambda$

$a = First(\beta_k' \dots \beta_1' \#)$,

deci $A \to \alpha.\beta; a$ valida pentru prefixul viabil $\gamma = \lambda$.

$n \longrightarrow n+1$: Fie $\gamma = X_1 \dots X_{n+1}, |\gamma| = n+1$ și $A \to \alpha.\beta; a \in goto(I_0, X_1 \dots X_{n+1})$.

Notăm cu $I_t = goto(I_0, X_1 \dots X_n), I_j = goto(I_0, X_1 \dots X_{n+1})$, deci $I_j = goto(I_t, X_{n+1})$. Avem deci $A \to \alpha.\beta; a \in goto(I_t, X_{n+1})$. Rezultă că există $B \to \alpha'.X_{n+1}\beta'; b \in I_t$ astfel încât

$B \to \alpha' X_{n+1}.\beta'; b \in I_j, \qquad \beta' = B_1\gamma_1, \qquad b = b_1$

$B_1 \to B_2\gamma \in P, B_1 \to B_2\gamma_2 \in I_j, \qquad b_2 \in First((\gamma_1 b_1) = First(\gamma_1 b)$

$\vdots$

$B_{k-1} \to \gamma_k \in P, B_{k-1} \to .\gamma_k; b_k \in I_j,\ b_k \in First(\gamma_{k-1} \dots \gamma_1 b)$.

Există $\gamma_{k-1}', \dots, \gamma_1' \in \Sigma^*$, $\gamma_{k-1} \underset{d}{\overset{*}{\Rightarrow}} \gamma_{k-1}', \dots, \gamma_1 \underset{d}{\overset{*}{\Rightarrow}} \gamma_1'$ și $\gamma_{k-1} \dots \gamma_1 b \underset{d}{\overset{*}{\Rightarrow}} \gamma_{k-1}' \dots \gamma_1' b$, $b_k = First(\gamma_{k-1}' \dots \gamma_1' b)$. Avem posibilitățile:

a) $A \to \alpha.\beta; a = B \to \alpha' X_{n+1}.\beta'; b \in I_t$
b) $A \to \alpha.\beta; a = B_{k-1} \to .\gamma_k; First(\gamma_{k-1}' \dots \gamma_1' b)$

Din $B \to \alpha'.X_{n+1}\beta'; b \in I_t = goto(I_0, X_1 \dots X_n)$, din ipoteza de inducție rezultă că $B \to \alpha'.X_{n+1}\beta'; b$ validă pentru prefixul $X_1 \dots X_n$, deci:

i. $S \underset{d}{\overset{*}{\Rightarrow}} \delta B w \underset{d}{\Rightarrow} \delta\alpha' X_{n+1}\beta' w$
ii. $\delta\alpha' = X_1 \dots X_n$
iii. $b = First(w\#)$.

În cazul a), din ii. rezulta $\delta\alpha' X_{n+1} = X_1 \dots X_n X_{n+1}$ și deci $A \to \alpha.\beta; a = B \to \alpha' X_{n+1}.\beta'; b$ validă pentru prefixul viabil $\gamma = X_1 \dots X_n X_{n+1}$.

În cazul b), avem:

- $S \underset{d}{\overset{*}{\Rightarrow}} \delta B w \underset{d}{\Rightarrow} \delta\alpha' X_{n+1}\beta' w = \delta\alpha' X_{n+1} B_1 \gamma_1 w \underset{d}{\overset{*}{\Rightarrow}} \delta\alpha' X_{n+1} B_1 \gamma_1' w \underset{d}{\overset{*}{\Rightarrow}} \dots \underset{d}{\overset{*}{\Rightarrow}}$

  $\delta\alpha' X_{n+1} \gamma_k \gamma_{k-1}' \dots \gamma_1' w$
- $\delta\alpha' X_{n+1} = X_1 \dots X_n X_{n+1}$
- $a = b_k = First(\gamma_{k-1}' \dots \gamma_1' b) = First(\gamma_{k-1}' \dots \gamma_1' w\#)$

adica $A \to \alpha.\beta; a = B_{k-1} \to .\gamma_k; First(\gamma'_{k-1} \dots \gamma'_1 w\#)$ valida pentru prefixul viabil $\gamma = X_1 \dots X_n X_{n+1}$.

" $\Longleftarrow$ " Demonstram următoarea

Lemă. Dacă $S \overset{n}{\underset{d}{\Rightarrow}} \gamma A x$, atunci $\forall A \to \alpha \in P, A \to .\alpha; First(x\#) \in goto(I_0, \gamma)$.

Facem inducție după n.

$n = 0$: $S = \gamma A x, \gamma = x = \lambda, A = S$. Fie $S \to \alpha \in P$. Atunci, din definiția lui $closure\ \ S \to .\alpha; \# \in goto(I_0, \lambda) = I_0 = closure(\{S' \to .S; \#\})$.

$n \to n + 1$: $S \overset{n}{\underset{d}{\Rightarrow}} \gamma A x \underset{d}{\Rightarrow} \gamma\gamma_1 B x_1 x$. Din ipoteza de inductie:

$A \to \gamma_1 B x_1, First(x\#) \in goto(I_0, \gamma)$, deci $A \to \gamma_1 . B x_1, First(x\#) \in goto(I_0, \gamma\gamma_1)$ și atunci $\forall B \to \beta \in P, B \to .\beta; First(x_1 x\#) \in goto(I_0, \gamma\gamma_1)$ și cu aceasta am demonstrat Lema.

Fie acum $A \to \alpha.\beta; a$ validă pentru prefixul viabil γ. Aceasta înseamnă:

i. $\exists S \overset{*}{\underset{d}{\Rightarrow}} \theta A x \underset{d}{\Rightarrow} \theta\alpha\beta x$
ii. $\theta\alpha = \gamma$
iii. $a = First(x\#)$

Din Lema de mai sus rezultă că $A \to .\alpha\beta; First(x\#) \in goto(I_0, \theta)$, deci $A \to \alpha.\beta; First(x\#) \in goto(I_0, \theta\alpha)$, adică $A \to \alpha.\beta; a \in goto(I_0, \gamma)$.

Teorema 2. Gramatica independentă de context $G = (N, \Sigma, S, P)$ este $LR(1)$ dacă și numai dacă tabela $action$ construită cu algoritmul $LR(1)$ nu are intrări multiple.

Demonstrație. Fie G' extensia lui G și $C_G = \{I_0, \dots, I_n\}$ mulțimile canonice $LR(1)$ ale lui $G, I_0 = closure(\{S' \to S.; \#\})$.

" $\Longrightarrow$ " Fie G de tip $LR(1)$. Presupunem că tabela $action$ asociată are intrări multiple. Avem cazurile:

a) $\exists i, j \in \{0,1, \dots, n\}, a \in \Sigma,\ action[i, a] = shift\ j, action[i, a] = reduce\ B \to \gamma$ (conflict deplasare-reducere). Aceasta înseamnă că $\exists A \to \alpha.a\beta; b \in I_i, goto(I_i, a) = I_j$ și $\exists B \to \gamma.; a \in I_i$. Fie $\gamma_1 \in (N \cup \Sigma)^*$ astfel încât $I_i = goto(I_0, \gamma_1)$. Atunci, conform Teoremei 1, rezultă că $A \rightarrow \alpha.a\beta; b$ și $B \rightarrow \gamma.; a$ sunt valide pentru prefixul viabil γ_1. Atunci:

i. $\exists S \overset{*}{\underset{d}{\Rightarrow}} \theta A x \underset{d}{\Rightarrow} \theta\alpha a\beta x$

ii. $\theta\alpha = \gamma_1$

iii. $b = First(x\#)$

iv. $\exists S \overset{*}{\underset{d}{\Rightarrow}} \theta' B x' \underset{d}{\Rightarrow} \theta'\gamma x'$

v. $\theta'\gamma = \gamma_1$

vi. $a = First(x'\#)$

Avem:

$S \overset{*}{\underset{d}{\Rightarrow}} \theta' B x' \underset{d}{\Rightarrow} \theta'\gamma x' = \gamma_1 x'$

$S \overset{*}{\underset{d}{\Rightarrow}} \theta A x \underset{d}{\Rightarrow} \theta\alpha a\beta x = \gamma_1 a\beta x$

$First(a\beta x) = First(x') = a$

Cum G este $LR(1)$, rezulta $\theta = \theta', A = B, \gamma = \alpha a\beta$. Rezulta din ii., v. și din $\theta = \theta'$ că $\alpha = \gamma$, dar atunci $\lambda = a\beta$, contradicție, deoarece $a \in \Sigma$.

b) $action[i, a] = reduce\ A \rightarrow \alpha, action[i, a] = reduce\ B \rightarrow \beta$, pentru $0 \leq i \leq n, a \in \Sigma \cup \{\#\}, A \rightarrow \alpha \neq B \rightarrow \beta$ (conflict reducere-reducere). Atunci $A \rightarrow \alpha.; a \in I_i, B \rightarrow \beta.; a \in I_i$, $I_i = goto(I_0, \gamma), \gamma \in (N \cup \Sigma)^*$ prefix viabil.

Avem:

vii. $\exists S \overset{*}{\underset{d}{\Rightarrow}} \theta A x \underset{d}{\Rightarrow} \theta\alpha x$

viii. $\theta\alpha = \gamma$

ix. $a = First(x\#)$

x. $\exists S \overset{*}{\underset{d}{\Rightarrow}} \theta' B x' \underset{d}{\Rightarrow} \theta'\beta x'$

xi. $\theta'\beta = \gamma$

xii. $a = First(x'\#)$

Rezultă:

$First(x) = First(x') = a, S\overset{*}{\underset{d}{\Rightarrow}}\theta Ax \underset{d}{\Rightarrow} \theta\alpha x = \gamma x, S\overset{*}{\underset{d}{\Rightarrow}}\theta' Bx' \underset{d}{\Rightarrow} \theta'\beta x' = \gamma x'$.

Cum G este $LR(1)$, rezulta ca $\theta = \theta', A = B, \alpha = \beta$, deci $A \to \alpha = B \to \beta$, contradictie.

" $\Leftarrow$ " Fie G astfel ca $action$ nu are intrări multiple. Presupunem că G nu este $LR(1)$.

Atunci există două derivări drepte:

$S \overset{*}{\underset{d}{\Rightarrow}} \alpha Au \underset{d}{\Rightarrow} \alpha\beta u = \gamma u, u \in \Sigma^*$ și $S \overset{*}{\underset{d}{\Rightarrow}} \alpha' A' u' \underset{d}{\Rightarrow} \alpha'\beta' u' = \alpha\beta v = \gamma v, v \in \Sigma^*$
astfel încât $FIRST_k(u) = FIRST_k(v)$, **pentru care cel puțin una din relațiile** $\alpha \neq \alpha', A \neq A', \beta \neq \beta'$ este adevărată.
În conformitate cu Lema anterioară, rezultă că:
$A \to .\beta; First(u\#) \in goto(I_0, \alpha)$ (1)
$A' \to .\beta'; First(u'\#) \in goto(I_0, \alpha')$ (2)
$A \to \beta.; First(u\#) \in goto(I_0, \alpha\beta) = goto(I_0, \gamma)$ (3)
$A' \to \beta'; First(u'\#) \in goto(I_0, \alpha'\beta')$ (4)
Avem cazurile:

c) $u' = v$. Atunci $\alpha\beta = \alpha'\beta' = \gamma$, $First(u') = First(v) = First(u)$, deci $First(u\#) = First(u'\#) = x$. Fie $I_j = goto(I_0, \gamma), 0 \leq j \leq n$. Atunci: $action[j, x] = (\beta, r_1), action[j, x] = (\beta', r_2)$, unde $r_1: A \to \beta, r_2: A' \to \beta'$.
Daca $\alpha = \alpha'$, atunci din $\alpha\beta = \alpha'\beta' = \gamma$, rezulta ca $\beta = \beta'$. Cum una dintre relatiile $\alpha \neq \alpha', A \neq A', \beta \neq \beta'$ este adevărată, atunci $A \neq A'$, deci $r_1 \neq r_2$.
Daca $\alpha \neq \alpha'$, atunci din $\alpha\beta = \alpha'\beta' = \gamma$ rezulta ca $\beta \neq \beta'$, deci din nou $r_1 \neq r_2$.
Atunci $action$ are intrari multiple, contradictie.

d) $|u'| < |v|$. Rezulta ca $v = v'u'$. Avem $\alpha'\beta'u' = \alpha\beta v$, deci $\alpha'\beta'u' = \alpha\beta v'u'$, adica $\alpha'\beta' = \alpha\beta v'$ (5)
Totodata $First(v) = First(v') = First(u) = x \in \Sigma$ (6)
d1) $\alpha' = \alpha$. Ținând cont și de (5), rezultă că $\beta' = \beta v'$. Atunci:
$A' \to .\beta'; First(u'\#) = A' \to .\beta v'; First(u'\#) \in goto(I_0, \alpha')$, deci $A' \to \beta.v'; First(u'\#) \in goto(I_0, \alpha'\beta) = goto(I_0, \alpha\beta) = goto(I_0, \gamma)$ (7)
Din (3) si (7) rezulta ca $action[j, x] = reduce\ A \to \beta, action[j, x] = shift\ k$ unde $I_k = goto(I_j, x)$, deci $action$ are intrari multiple, contradictie.
d2) $|\alpha'| < |\alpha|, \alpha = \alpha'\alpha''$ (8)

Din (5) si (8), $\alpha'\beta' = \alpha'\alpha''\beta v' \Longrightarrow \beta' = \alpha''\beta v'$ (9)
Avem:
$A' \to .\beta'; First(u'\#) \in goto(I_0, \alpha') \Longrightarrow A' \to \alpha''\beta.v'; First(u'\#) \in goto(I_0, \alpha'\alpha''\beta) = goto(I_0, \alpha\beta) = goto(I_0, \gamma) = I_j$ (10)
Din (3), (6) si (10) rezulta ca $action[j, x] = reduce\ A \to \beta, action[j, x] = shift\ k$ unde $I_k = goto(I_j, x)$, deci $action$ are intrari multiple, contradictie.
d3) $|\alpha'| > |\alpha|, \alpha' = \alpha\,\alpha''$ (11)
Din (5) si (11) rezulta $\alpha\alpha''\beta' = \alpha\beta v' \Longrightarrow \alpha''\beta' = \beta v'$ (12)

d3.1)Presupunem β prefix propriu al lui α'', adica $\alpha'' = \beta\alpha'''$. Din $\alpha' = \alpha\alpha'' = \alpha\beta\alpha'''$ si $\alpha'\beta' = \alpha\beta v'$ rezulta $\alpha\beta\alpha'''\beta' = \alpha\beta v' \Longrightarrow \alpha'''\beta' = v' \in \Sigma^*, First(v') = First(\alpha'''\beta') \Longrightarrow First(v') = First(v) = First(\alpha''')$. Avem:
$A' \to .\beta'; First(u'\#) \in goto(I_0, \alpha\beta\alpha''')$ (13)
$A \to .\beta; First(u\#) \in goto(I_0, \alpha\beta)$ (14)
Din (13) și ținând cont că $\alpha''' \in \Sigma^+$, rezulta ca exista $B \to \delta.\alpha'''A\theta; First(u'\#) \in goto(I_0, \alpha\beta) = goto(I_0, \gamma)$ (15)
Din (3) si (15) rezulta ca $action[j, x] = reduce\ A \to \beta, action[j, x] = shift\ k$ unde $I_k = goto(I_j, x)$, deci $action$ are intrari multiple, contradictie.
d3.2) Daca $\beta = \alpha''$, adica $\alpha' = \alpha\beta = \gamma = \alpha'\beta' \Longrightarrow \beta' = \lambda$. Cum $\alpha'\beta'u' = \alpha\beta v$, rezulta $u' = v \in \Sigma^* \Longrightarrow First(u'\#) = First(v\#)$ (16)
Din (3), (4) si (16) rezulta $action[j, x] = reduce\ A \to \beta,\ \ action[j, x] = reduce\ A' \to \beta'$. Deoarece $\beta' = \lambda, \beta \neq \lambda$, rezulta $action$ are intrari multiple, contradictie.
d3.3) Presupunem α'' prefix propriu al lui β, deci $\beta = \alpha''\alpha'''$ (17)
Din (11), (12), (17) ($\alpha' = \alpha\alpha'', \alpha''\beta' = \beta v', \beta = \alpha''\alpha'''$) rezulta ca $\alpha''\beta' = \beta v' = \alpha''\alpha'''v' \Longrightarrow \beta' = \alpha'''v'$. Ținând cont și de (3), (4) obținem:
$A \to \beta.; First(u\#) \in goto(I_0, \alpha\beta)$
$A' \to \alpha'''.v'; First(u'\#) \in goto(I_0, \alpha'\alpha''')$
Dar $\alpha'\alpha''' = \alpha\alpha''\alpha''' = \alpha\beta \Longrightarrow A' \to \alpha'''.v'; First(u'\#) \in goto(I_0, \alpha\beta)$ (18)
Din (3) si (18) rezulta ca $action[j, x] = reduce\ A \to \beta, action[j, x] = shift\ k$ unde $I_k = goto(I_j, x)$, deci $action$ are intrari multiple, contradictie.

e) $|u'| > |v|$ Analog cazului d).

Teorema 3. Orice gramatică de tip $LR(k), k \geq 2$, este echivalentă cu o gramatică de tip $LR(1)$.

Pentru algoritmii de tip LR este suficient $k = 1$, adică se ia în considerare doar un singur simbol lookahead.

Teorema 4. Orice gramatică de tip $LL(k), k \geq 0$, este echivalentă cu o gramatică de tip $LR(k)$.