

# Gramatici LR(k)

- analizoare LR(k)
    - analiza sintactica ascendenta
    - secv. de intrare este citita de la stanga spre dreapta
    - se folosesc derivari de dreapta
- metoda: deplasare - reducere

# Gramatica LR(K)

Analizoarele sintactice LR(k) lucreaza cu gramatica imbogatita:

$$G' = (N \cup \{S'\}, \Sigma, P \cup \{S' \rightarrow S\}, S')$$

$$(S' \notin N)$$

pentru a evita ca simbolul de start sa apara in membrul drept al unei reguli de productie.

# Gramatica LR(K)

O gramatica  $G = (N, \Sigma, P, S)$

este de tip LR(k) pentru  $k \geq 0$

daca din:

- $S \xRightarrow{*}_{dr} \alpha A w \Rightarrow_{dr} \alpha \beta w$
- $S \xRightarrow{*}_{dr} \gamma B x \Rightarrow_{dr} \alpha \beta y$
- $FIRST_k(w) = FIRST_k(y)$

rezulta ca:

- $A = B$
- $x = y$
- $\alpha = \gamma$

# Definitie: Gramatica LR(K)

Daca  $S$  – nu apare in m.d. al unei r.p.:

si daca:

- $S \overset{*}{\Rightarrow}_{dr} \alpha A w \Rightarrow_{dr} \alpha \beta w$
- $S \overset{*}{\Rightarrow}_{dr} \gamma B x \Rightarrow_{dr} \alpha \beta y$
- $FIRST_k(w) = FIRST_k(y)$

atunci:

- $A = B$
- $x = y$
- $\alpha = \gamma$

# Gramatici LR(K) - terminologie

## Prefix viabil

Fie:  $S \overset{*}{>}_{\text{dr}} \alpha A w \Rightarrow_{\text{dr}} \alpha \beta w$

Orice prefix al lui  $\alpha\beta$  se numeste prefix viabil

## Element de analiza LR(k)

se defineste ca fiind:  $[A \rightarrow \alpha.\beta, u]$

unde  $A \rightarrow \alpha\beta \in P$  si  $u \in \Sigma^k$

- descrie stadiul in care, avand in vedere regula de productie  $A \rightarrow \alpha\beta$ , a fost detectat  $\alpha$  (e in varful stivei) si se asteapta sa fie detectat  $\beta$ :

- $S \overset{*}{>}_{\text{dr}} \gamma A w \Rightarrow_{\text{dr}} \gamma \alpha \beta w$

- $u = \text{FIRST}_k(w)$

valabil pentru prefixul viabil  $\gamma\alpha$

# Analizor sintactic LR(K)

- la multimea cuv. de analizat se adauga la sfarsit \$
  - \$ - marcator de sfarsit de cuvant
- gramatica imbogatita
- constructia colectiei canonice
- constructia tabelului de analiza
- analiza: automat

# Colectia canonica LR(K)

$C = \{I_i\text{-elementele de analiza pentru un prefix viabil}\}$

- in  $I_0$  avem un prim element de analiza
- am un element din  $I_j$  *(pentru fiecare)*  
 $\Rightarrow$  le construiesc pe celelalte: functia *Closure*
- am o multime  $I_j$  *(pentru fiecare)*  
 $\Rightarrow$  construiesc multimele *goto*( $I_i, X$ )

Terminologie:  $I_i$  – stare a automatului

Notatie:  $\mathcal{E}$  – multimea elementelor de analiza

# Constructia colectiei canonice LR(k)

$C = \{ I_i \text{-elementele de analiza pentru un prefix viabil} \}$

in  $I_0$  avem:  $[S' \rightarrow .S, \dots]$

- $I_0 = \text{Closure}([S' \rightarrow .S, \dots])$
  - $C = \{ I_0 \}$
  - repeta
    - pentru toti  $I_i$  din  $C$ ,  $X \in (N \cup \Sigma)$  executa
$$C = C \cup \text{goto}(I_i, X)$$
    - sf. pentru
- pana cand  $C$  nu se mai modifica



**K=0:            LR(0)**

# Gramatica imbogatita

- se adauga  $S'$ 
  - nou simbol de start
  - $S' \rightarrow S$

# Funcția *Closure*

# LR(0)

- $Closure : \text{Part}(\mathcal{E}) \rightarrow \text{Part}(\mathcal{E})$

- Fie:  $e \in \mathcal{E}$

daca  $e = [A \rightarrow \alpha . B\beta]$

atunci  $\forall B \rightarrow \delta \in P: [B \rightarrow . \delta] \in Closure(e)$

# Funcția *goto* LR(0)

- $goto : \text{Part}(\mathcal{E}) \times (\mathbf{N} \cup \Sigma) \rightarrow \text{Part}(\mathcal{E})$
- $goto(I, X) = \text{Closure}(\{[A \rightarrow \alpha X \beta] \mid [A \rightarrow \alpha.X\beta] \in I\})$

# Tabelul de analiza LR(0)

		goto		
		N	U	$\Sigma$
I <sub>0</sub> I <sub>1</sub> : :	actiune			
	deplasare (s)			
	reducere (nr. r.p.)			
	acceptare (acc)			
	eroare			

- $T(I_i, \text{actiune}) =$ 
    - s (shift, deplasare)
      - daca:  $[A \rightarrow \alpha.\beta] \in I_i$  ,  $\beta \neq \epsilon$
      - si:  $T(I_i, X) = I_j$ , daca  $I_j = \text{goto}(I_i, X)$
    - L (reducere cu r.p. nr. L)
      - daca  $[A \rightarrow \alpha.] \in I_i$ 
        - $A \rightarrow \alpha \in P$  : regula de prod. cu numarul L
        - si:  $T(I_i, X)$  nu se completeaza
    - acc      daca:  $[S' \rightarrow S.] \in I_i$
- Toate celelalte cazuri se considera eroare .

## Tabelul de analiza LR(0)

# Automatul LR(0) – model matematic

- configuratie:  
 $(\alpha, \beta, \Pi)$   
(stiva\_de\_lucru, banda\_de\_intrare, banda\_de\_iesire)
- pe stiva: prefixe viabile, stari ale analizorului
- config. initiala:  $(\$0, w$,  $\varepsilon$ )$
- config. finala:  $(\$0S I_{acc}, \$, \Pi)$

# Tranzitii

- deplasare:

$$(\$ \gamma s_k, a_i..a_n \$, \Pi) \vdash (\$ \gamma s_k a_i s_m, a_{i+1}...a_n \$, \Pi)$$

daca:  $T(s_k, \text{actiune}) = s$  si  $T(s_k, a_i) = s_m$

- reducere

$$(\$ \gamma s_{p-1} X_p s_p \dots X_k s_k, a_i..a_n \$, \Pi) \vdash (\$ \gamma s_{p-1} A s_m, a_i...a_n \$, L\Pi)$$

daca:  $T(s_k, \text{action}) = L$

si:  $A \rightarrow X_p \dots X_k - \text{r.p. cu nr. } L$

$T(s_{p-1}, \text{actiune}) = s$

$T(s_{p-1}, A) = s_m$

- acceptare:  $(\$ 0 S s_{\text{acc}}, \$, \Pi) \vdash \text{acc.}$

- eroare: *orice alta situatie*

# Automatul LR(0) – model matematic



- Gramatica data prin urmatoarele r.p. este LR(0) ?

$$S \rightarrow Ax$$

$$S \rightarrow By$$

$$A \rightarrow a$$

$$B \rightarrow a$$

**K=1: SLR, LR(1), LALR**

# Analizor sintactic SLR

- SLR = Simple LR
- element de analiza SLR:  
     $[A \rightarrow \alpha.\beta, u]$   
     $u = \text{FOLLOW}_1(A)$   
     $|u| = 1$
- SLR: tine cont de predictie numai pentru reducere

# Analizor sintactic SLR

- constructia colectiei canonice ( $\sim LR(0)$ )
  - $[A \rightarrow \alpha.\beta, u]$  ,  $u = FOLLOW_1(A)$
- constructia tabelului de analiza SLR
  - actiunea de reducere depinde de predictia  $u$   
 $\Rightarrow$ reducerea va avea o coloana pentru fiecare  $a \in \Sigma$
  - tabelul: linii: elementele colectiei canonice  
coloane:  $N \cup \Sigma \cup \{\$ \}$   
celula:  $s_{stare}, r_{nr.r.p}, acc$
- analizorul  $\sim$  analizorul pt.  $LR(0)$ 
  - automat: configuratii si tranzitii

# Tabelul de analiza

actiune: **reducere**

+

**deplasare**

(goto)

$X \in \Sigma \cup \{\$ \}$

$X \in N \cup \Sigma$

linii: elementele colectiei canonice

coloane:  $N \cup \Sigma \cup \{\$ \}$

# Construirea tab. de analiza LR(1)

- $[A \rightarrow \alpha.X\beta, b] \in I_i$  :  $goto(I_i, X) = I_j \Leftarrow$  functia *goto*  
 $action(I_i, X) = sj$
- $[A \rightarrow \alpha. , a] \in I_i$   $action(I_i, a) = rL$   
L – nr. reg. de productie:  $A \rightarrow \alpha$   
 $A \triangleleft S'$
- $[S' \rightarrow S., \$] \in I_i$   $action(I_i, \$) = acc$

Obs: o gram. este LR\* daca tabelul de analiza nu  
contine conflicte; si reciproc

# Analizor sintactic LR(1)

- îmbogățirea gramaticii
  - construcția colecției canonice
- element de analiza LR(1):
- $[A \rightarrow \alpha.\beta, u], |u| = 1$
- construcția tabelului de analiza
  - analiza: automat

# Colectia canonica LR(1)

- elem. initial

$$[S' \rightarrow .S, \$]$$

- *Closure*

$$[A \rightarrow \alpha.B\beta, a] \Rightarrow [B \rightarrow .\gamma, b] \in \text{Closure}([A \rightarrow \alpha.B\beta, a])$$

$$B \rightarrow \gamma \quad \forall b \in \text{FIRST}_1(\beta a)$$

- *goto*

$$\text{goto}(I, X) =$$

$$\text{Closure}(\{[A \rightarrow \alpha X.\beta, a] \mid [A \rightarrow \alpha.X\beta, a] \in I\})$$





- analizorul LR(1)
- analiza LR(1)

# Analizor sintactic LALR

- $[ \underbrace{A \rightarrow \alpha.\beta}_{\text{nucleu}} , a ]$
- colectia canonica LR(1)
- fuzioneaza elementele de analiza cu nuclee identice si care nu creeaza conflicte
- predictia: reuniunea predictiilor

# LR (1 –uri)

- Conflict:

$[ A \rightarrow \alpha_1 . a \alpha_2 , u ]$

$[ B \rightarrow \beta_1 . , a ]$

$[ A \rightarrow \alpha_1 . , a ]$

$[ B \rightarrow \beta_1 . , a ]$