

# SLR(1) - analyzátory

Ing. Viliam Hromada, PhD.

C-510  
Ústav informatiky a matematiky  
FEI STU

`viliam.hromada@stuba.sk`



## Syntaktický analyzátor jazyka SLR(1)

- Silnejším a v praxi použiteľnejším analyzátorom je Simple LR(1), skrátene SLR(1).
- Konštrukcia vychádza z LR(0) automatu, avšak v prípade, že niektorý z jeho stavov obsahuje konflikt, tak sa ho pokúsi vyriešiť na základe aktuálneho vstupného symbolu  $t$ .
- Konflikty vznikajú len, ak je signalizovaná redukcia.
- Preto je dobré skontrolovať, či má signalizovaná redukcia v danej gramatike zmysel vzhľadom na možné vetné formy a vstupný symbol.



## Hlavná myšlienka konštrukcie SLR(1)

- Nech je v gramatike pravidlo  $B \rightarrow \rho$ , na vstupe reťazec  $tw$  a na vrchu zásobníka pravá strana pravidla  $\rho$ .  
 $B \in N, \rho \in (N \cup T)^*, t \in T_\epsilon, w \in T^*$ .
- Redukcia podľa  $B \rightarrow \rho$  má **zmysel len vtedy**, ak  $t \in FOLLOW(B)$ , t.j. ak  $\dots Btw$  môže byť nejaká **pravá vetná forma**.

Zásobník $\square$	Zvyšok vstupu:	
$\dots\rho$	$tw$	pred redukciou podľa $B \rightarrow \rho$
$\dots B$	$tw$	po redukcii podľa $B \rightarrow \rho$

## Konštrukcia tabuľky ACTION SLR(1)

1. Ku gramatike  $G = (N, T, P, S)$  pridať nový počiatkový neterminál  $S'$  a skonštruovať  $LR(0)$ -automat s počiatkovým stavom  $s_0 = CLOSURE_0(\{S' \rightarrow \bullet S\})$ .
2. Určiť množiny  $FOLLOW(A)$  pre  $A \in N$ .
3. Ku všetkým položkám  $LR(0)$ -automatu tvaru  $B \rightarrow \rho \bullet$  priradiť symboly z množiny  $FOLLOW(B)$ .
4. Skonštruovať  $ACTION$  pre SLR(1) analyzátor nasledovne:
  - Položka  $S' \rightarrow S \bullet, \varepsilon$  v stave  $s$  signalizuje akceptáciu, t.j.  $A \in ACTION[s, \varepsilon]$ .
  - Položky  $B \rightarrow \rho \bullet, t$ , kde  $t \in FOLLOW(B)$  v stave  $s$  signalizujú redukciu podľa  $B \rightarrow \rho$ , ak sa na **vstupe nachádza**  $t$ , t.j.  $R_{B \rightarrow \rho} \in ACTION[s, t]$ .
  - Položky  $A \rightarrow \alpha \bullet t \beta$  v stave  $s$  signalizujú presun, ak sa na vstupe nachádza  $t$ , t.j.  $P \in ACTION[s, t]$ .



## Syntaktický analyzátor SLR(1)

- Tabuľka *GOTO* sa konštruuje rovnako, ako v prípade *LR(0)*.
- V prípade, že každá položka tabuľky *ACTION* SLR(1) analyzátoru obsahuje max. 1 akciu, gramatika sa nazýva *SLR(1)*-gramatika.
- Platí, že  $LR(0) \subset SLR(1)$ , t.j. niektoré gramatiky, ktoré neboli *LR(0)* (tabuľka *ACTION* obsahovala konflikty), sú *SLR(1)* gramatiky, (táto verzia tabuľky *ACTION* už konflikty neobsahuje).



## Príklad

Gramatika, ktorá je  $SLR(1)$ , ale nie je  $LR(0)$ :

$\langle \text{program} \rangle \rightarrow \mathbf{begin} \langle \text{príkazy} \rangle \mathbf{end}$

$\langle \text{príkazy} \rangle \rightarrow \langle \text{príkaz} \rangle ; \langle \text{príkazy} \rangle$

$\langle \text{príkazy} \rangle \rightarrow \varepsilon$

$\langle \text{príkaz} \rangle \rightarrow \mathbf{p}$



## Príklad

LR(0) automat doplnený o symboly množiny FOLLOW pri položkách signalizujúcich redukciu:

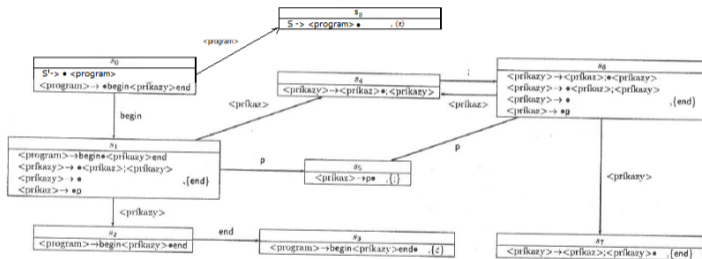


Figure: Prevezaté z Dederá, L.: Počítačové jazyky a ich spracovanie

## Príklad

<i>ACTION</i>	$s_0$	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
<b>begin</b>	P								
<b>end</b>		R3	P				R3	R2	
<b>;</b>					P	R4			
<b>p</b>		P					P		
$\epsilon$				R1					A

<i>GOTO</i>	$s_0$	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
<b>begin</b>	$s_1$								
<b>end</b>			$s_3$						
<b>;</b>					$s_6$				
<b>p</b>		$s_5$					$s_5$		
<program>	$s_8$								
<príkazy>		$s_2$					$s_7$		
<príkaz>		$s_4$					$s_4$		





## Konflikty syntaktického analyzátoru SLR(1)

- Ako  $LR(0)$ -analyzáto­ry, aj  $SLR(1)$ -analyzáto­ry sa nemusia dať zos­tro­jiť pre všetky gramatiky.  $SLR(1)$ -analyzáto­ry totižto môžu mať 2 typy konfliktov:
  1. Konflikt **redukcia-redukcia** - ak existujú 2 rôzne položky signalizujúce redukciu so spoločným očakávaným symbolom (symbolmi) v tom istom stave, napr.  
$$E \rightarrow E + T\bullet, \{id, \dots\}$$
$$E \rightarrow T\bullet, \{id, \dots\}$$
  2. Konflikt **presun-redukcia** - ak ten istý symbol signalizuje presun a zároveň je očakávaným symbolom pri redukcii v tom istom stave, napr.  
$$T \rightarrow (E)\bullet, \{id, \dots\}$$
$$E \rightarrow \bullet id$$



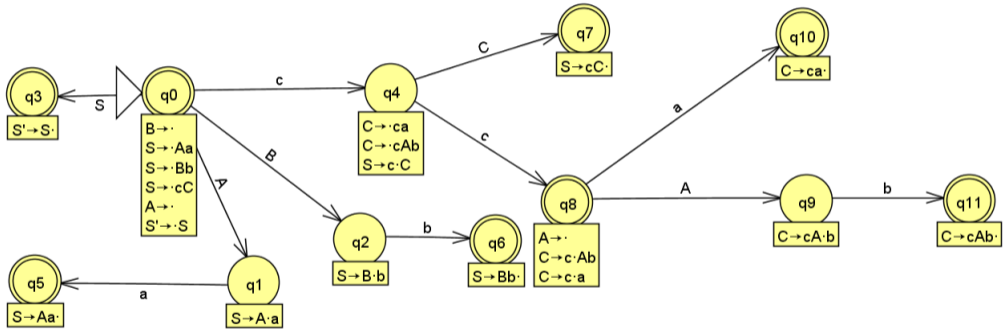
## Príklad - Konflikty syntaktického analyzátoru SLR(1)

Nech je daná gramatika  $G = (\{S, A, B, C\}, \{a, b, c\}, P, S)$  a pravidlá:

1.  $S \rightarrow Aa$
2.  $S \rightarrow Bb$
3.  $S \rightarrow cC$
4.  $A \rightarrow \varepsilon$
5.  $B \rightarrow \varepsilon$
6.  $C \rightarrow cAb$
7.  $C \rightarrow ca$



# LR(0)-automat



## ACTION-tabuľka

ACTION	$q_0$	$q_1$	$q_2$	$q_3$	$q_4$	$q_5$	$q_6$	$q_7$	$q_8$	$q_9$	$q_{10}$	$q_{11}$
$a$	R4	P							P/R4			
$b$	R4/R5		P						R4	P		
$c$	P				P							
$\epsilon$				A		R1	R2	R3			R7	R6

Vidíme, že v stave  $q_0$  je konflikt redukcia-redukcia a v stave  $q_8$  je konflikt presun-redukcia. Gramatika sa teda **nedá spracovať** SLR(1) analyzátorom a gramatika teda **nie je** SLR(1) gramatikou.



## Použitá literatúra

Aho, A., Lam, M., Sethi, R., Ullman, J.: *Compilers: Principles, techniques and tools.*

Dedera, L': *Počítačové jazyky a ich spracovanie.*

Linz, P.: *An Introduction to Formal Languages and Automata.*

