

SLR(1), LALR(1) - analizatory

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 než $LR(0)$ 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 .
- V $LR(0)$ vznikajú konflikty 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, t.j. ρ .

$$B \in N, \rho \in (N \cup T)^*, t \in T_\varepsilon, w \in T^*.$$

- Redukcia podľa $B \rightarrow \rho$ má **zmysel len vtedy**, ak $t \in FOLLOW(B)$, t.j. ak ...*B*tw môže byť nejaká **pravá vetná forma**.

Zásobník ☐ Zvyšok vstupu:	
... ρ	tw pred redukciou podľa $B \rightarrow \rho$
... B	tw po redukcii podľa $B \rightarrow \rho$

Konštrukcia tabuľky ACTION SLR(1)

1. Ku gramatike $G = (N, T, P, S)$ pridať nový počiatočný neterminál S' a skonštruovať $LR(0)$ -automat s počiatočným 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 $LR(0)$:

<program> → begin<príkazy>end

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

<príkaz> → p



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

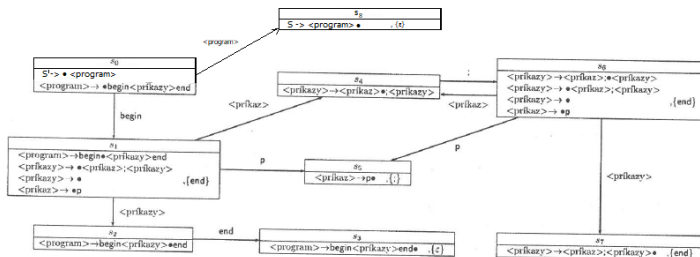


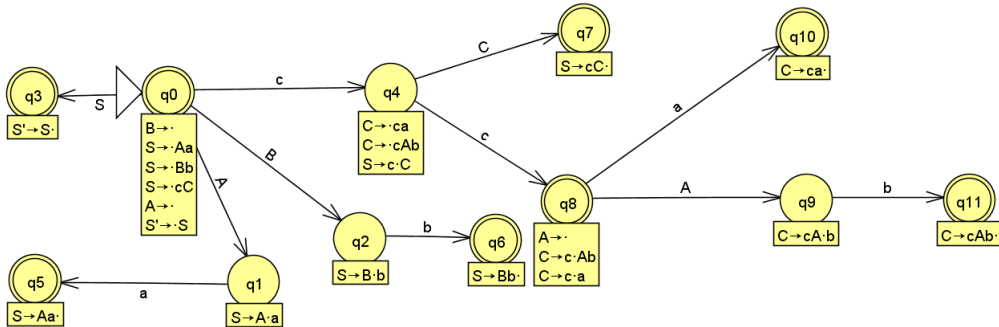
Figure: Prevzaté z Dederá, L.: Počítačové jazyky a ich spracovanie

Príklad

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

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

$LR(0)$ -automat



ACTION-tabul'ka

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							
ε				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.



Syntaktický analyzátor *LALR*(1)

- Vidíme, že gramatika zo slajdu 10 nebola $SLR(1)$ -gramatikou, pretože tabuľka *ACTION* obsahovala konflikty P/R a R/R.
- V tomto prípade sa teda analyzátor nedokázal jednoznačne rozhodnúť, akú akciu má vykonať, ani na základe aktuálneho vstupného symbolu a množín *FOLLOW*.
- Dôvod je ten, že $SLR(1)$ analyzátor si položí len jednoduchú otázku *ako vyzerajú vetné formy gramatiky*, ale nezamýšľa sa nad tým, **či majú vetné formy zmysel vzhľadom na aktuálnu situáciu s rozpoznanými pravými stranami pravidiel**.



Syntaktický analyzátor *LALR*(1)

- Vezmime napríklad stav q_0 v $SLR(1)$ -automate. Tento stav je tvorený položkami:
 - $S' \rightarrow \bullet S$
 - $S \rightarrow \bullet Aa$
 - $S \rightarrow \bullet Bb$
 - $S \rightarrow \bullet cC$
 - $A \rightarrow \bullet$
 - $B \rightarrow \bullet$
- Konflikt v stave q_0 vytvárajú položky $A \rightarrow \bullet$ a $B \rightarrow \bullet$ a symbol b na vstupe, keďže patrí aj do $FOLLOW(A)$, aj do $FOLLOW(B)$.
- Avšak položme si otázku - **má zmysel uvažovať pre položku $A \rightarrow \bullet$ vstupný symbol b ?**



Syntaktický analyzátor *LALR(1)*

- Je pravda, že naozaj $b \in FOLLOW(A)$. Avšak má táto situácia zmysel v stave q_0 ? Má z hľadiska **aktuálnej situácie rozpoznávania pravých strán** zmysel uvažovať, že by sa po redukcii dostal na vrch zásobníka neterminál A a na vstupe by bolo b ?
- Ako vlastne vznikla v stave q_0 položka $A \rightarrow \bullet$?
- Položka $A \rightarrow \bullet$ vznikla ako odpoveď na **požiadavku** rozpoznania neterminálu A v položke $S \rightarrow \bullet Aa$.
- Teda ak sa aj **rozpozná** neterminál A pomocou položky $A \rightarrow \bullet$, ako ďalší symbol by sa v položke $S \rightarrow A \bullet$ malo rozpoznať **a**!
- Teda v stave q_0 **nemá zmysel** uvažovať, že by pri redukcii podľa pravidla $A \rightarrow \bullet$ malo byť na vstupe b !



Syntaktický analyzátor *LALR*(1)

- Podobne vezmime stav q_8 , v ktorom sú položky:
 - $C \rightarrow c \bullet a$
 - $C \rightarrow c \bullet Ab$
 - $A \rightarrow \bullet$
- Konflikt v stave q_8 vytvárajú položky $A \rightarrow \bullet$ a $C \rightarrow c \bullet a$ a symbol a na vstupe, keďže $SLR(1)$ analyzátor si myslí, že a je možné alebo presúvať kvôli položke $C \rightarrow c \bullet a$, alebo podľa neho redukovať $A \rightarrow \bullet$, keďže $a \in FOLLOW(A)$.
- Avšak položíme si otázku - **má zmysel uvažovať pre položku $A \rightarrow \bullet$ vstupný symbol a pri redukcii ?**



Syntaktický analyzátor *LALR*(1)

- Je pravda, že naozaj $a \in FOLLOW(A)$. Avšak má táto situácia zmysel v stave q_8 ? Má z hľadiska **aktuálnej situácie rozpoznávania pravých strán** zmysel uvažovať, že by sa po redukcii dostal na vrch zásobníka neterminál A a na vstupe by bolo a ?
- Ako vlastne vznikla v stave q_8 položka $A \rightarrow \bullet$?
- Položka $A \rightarrow \bullet$ vznikla ako odpoveď na **požiadavku** rozpoznania neterminálu A v položke $C \rightarrow c \bullet Ab$.
- Teda ak sa aj **rozpozná** neterminál A pomocou položky $A \rightarrow \bullet$, ako ďalší symbol by sa v položke $C \rightarrow cA \bullet b$ malo rozpoznať **b**!
- Teda v stave q_8 **nemá zmysel** uvažovať, že by pri redukcii podľa pravidla $A \rightarrow \bullet$ malo byť na vstupe a ! Ak je na vstupe a , tak má zmysel ho **len presúvať** kvôli položke $C \rightarrow c \bullet a$.



Rozdiel $SLR(1)$ a $LALR(1)$

- Z uvedeného vyplýva, že zatiaľ čo $SLR(1)$ -analyzátor používa na rozhodovanie o akcii **elementárnu informáciu** o štruktúre gramatiky (vetné formy) **bez ohľadu na to**, v akom stave sa nachádza derivácia,
- $LALR(1)$ analyzátor si už kladie **detailnejšiu otázku** ohľadom toho **v akej situácii sa môže derivácia nachádzať**, resp. **ako sa vlastne dostali jednotlivé položky do stavov príslušného automatu** a **aký symbol by mal byť neskôr na vstupe** vzhľadom na rozpoznávanie **pravých strán pravidiel**.



LALR(1)

- $LALR(1)$ -analyzátor je teda **silnejším** typom LR -analyzátoru než $SLR(1)$, a teda logicky aj silnejším analyzátorom než $LR(0)$.
- $LR(0) \subset SLR(1) \subset LALR(1)$
- V praxi býva $LALR(1)$ **najčastejšou** voľbou syntaktického analyzátoru, spolu s $LL(1)$ -analyzátorom.
- https://en.wikipedia.org/wiki/Comparison_of_parser_generators.



LALR(1)

- $LALR(1)$ -analyzátor, tzv. *Look-Ahead Left-Right* analyzátor, teda používa 1 vstupný symbol na rozhodnutie o tom, akú akciu má vykonať.
- Tento vstupný symbol sa už neporovnáva so symbolmi z množiny *FOLLOW*, ako tomu je pri $SLR(1)$ -analyzátore, ale sa porovnáva s tzv. **očakávanými symbolmi**.
- Základom konštrukcie $LALR(1)$ -analyzátoru sú teda tzv. **očakávané symboly na vstupe**. Ide o symboly, ktoré analyzátor **očakáva na vstupe v momente**, keď je signalizovaná **redukcia** podľa nejakého pravidla.

LALR(1)

- Konštrukcia $LALR(1)$ -analyzátoru vychádza z konštrukcie tzv. $LALR(1)$ -automatu.
- $LALR(1)$ -automat vznikne tak, že sa najprv zostrojí štandardný $LR(0)$ -automat a následne sa k jednotlivým položkám pridajú **očakávané symboly**, čím sa z položiek $LR(0)$ -automatu vytvoria tzv. $LR(1)$ -položky.



$LALR(1)$ -automat

- Môže sa stať, že jedna $LR(1)$ -položka obsahuje viacero očakávaných symbolov.
- V takom prípade ju zapíšeme skrátene ako $A \rightarrow \alpha_1 \bullet \alpha_2, \{l_1, \dots, l_m\}$
- T.j. bola rozpoznaná časť α_1 pravidla $A \rightarrow \alpha_1 \alpha_2$, pričom po rozpoznaní celej pravej strany sa na vstupe očakáva niektorý symbol z množiny $\{l_1, \dots, l_m\}$.
- Pri konštrukcii $LALR(1)$ -automatu postupujeme obdobne, ako pri $LR(0)$, s tým rozdielom, že je potrebné doplniť k všetkým položkám očakávané symboly.

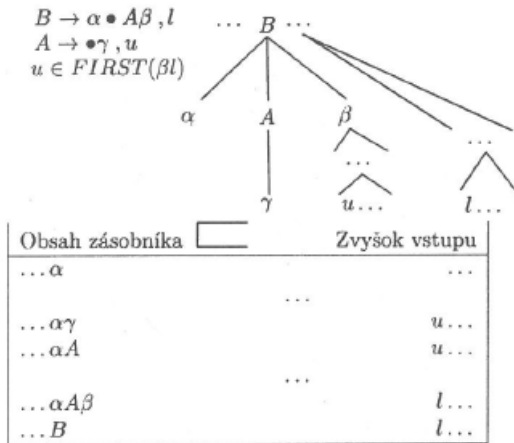
$LALR(1)$ -automat, dopĺňanie očkávaných symbolov

- Potrebujeme určiť, ako doplniť do stavov k jednotlivým položkám očakávané symboly.
- Každý stav môže obsahovať 2 typy položiek:
 1. *Základné* - do stavu sa dostanú presunom $\bullet$ za symbol, na ktorý sa prechod uskutočňuje; očakávané symboly sa len prenášajú.
 2. *Uzáverové* - určujú sa na základe operácie *CLOSURE*₁; Ak sa uzáver tvorí pre položku $B \rightarrow \alpha \bullet A\beta$, L_1 , $L_1 \subseteq T_\epsilon$, potom pridávame položky $A \rightarrow \bullet \gamma$, L_2 , kde $L_2 = FIRST(\beta I)$, $I \in L_1$.



Vysvetlenie uzáverovej operácie *CLOSURE*¹

- Uvažujme položku $B \rightarrow \alpha \bullet A\beta, l$

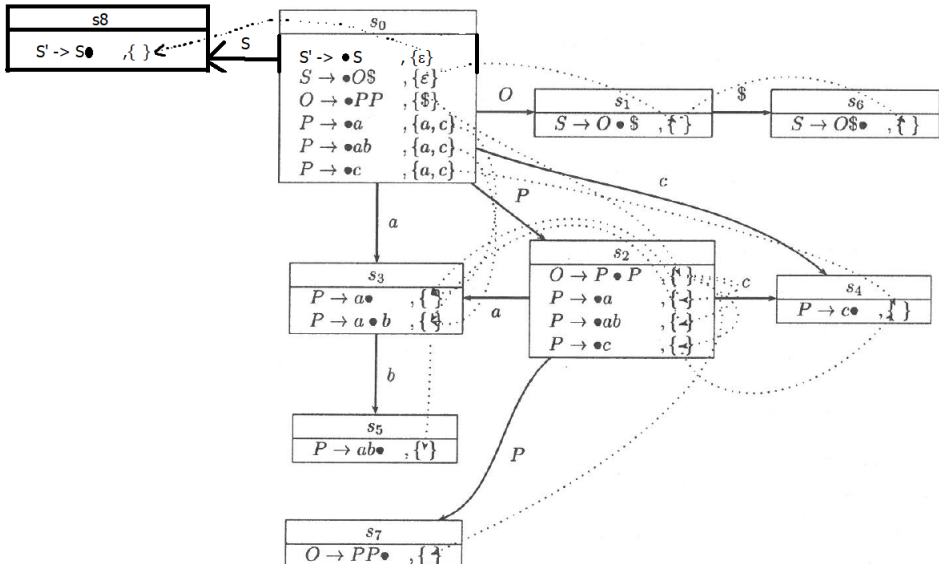


Vysvetlenie uzáveru položky $B \rightarrow \alpha \bullet A\beta, I$

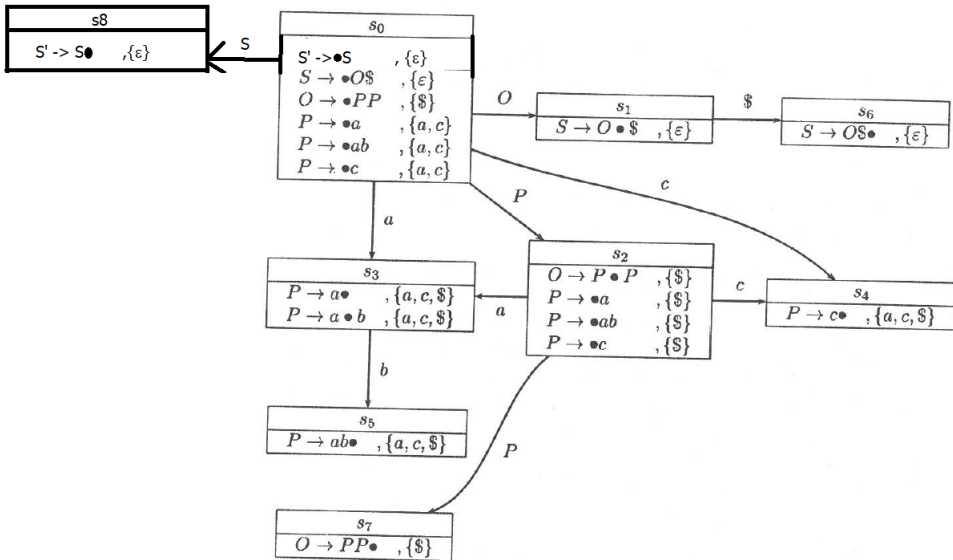
- Podobne ako pri $LR(0)$, ak sa má ako ďalší symbol na vrchu zásobníka rozpoznať A , potom je najprv potrebné rozpoznať niektoré z A -pravidiel (t.j. jeho pravú stranu), aby sa po jeho redukcii na vrch zásobníka dostalo práve A .
- V okamihu redukcie tohto A -pravidla musí byť na vstupe taký terminál, ktorým sa môžu začínať reťazce odvodené z časti β pravidla $B \rightarrow \alpha A \beta$, tzn. symbol z množiny $FIRST(\beta I)$.
- $FIRST(\beta I)$ preto, lebo ak β môže generovať/odvodiť ε , tak sa na vstupe môže očakávať aj symbol I .
- Teda ak je daná položka $B \rightarrow \alpha \bullet A \beta$, na základe ktorej vznikli v $LR(0)$ -automate v tom istom stave položky $A \rightarrow \bullet \gamma$, potom ak položka $B \rightarrow \alpha \bullet A \beta$ má očakávané symboly I , t.j. je daná $LR(1)$ položka $B \rightarrow \alpha \bullet A \beta, I$, tak položky $A \rightarrow \bullet \gamma$ majú očakávané symboly u , kde $u \in FIRST(\beta I)$.



Príklad šírenia symbolov



Príklad - výsledný *LALR*(1) automat



ACTION, GOTO pre LALR(1)-analyzátor

- Tabuľka *GOTO* sa zostrojí analogicky ako pri $LR(0)$, resp. $SLR(1)$ analyzátore.
- Tabuľka *ACTION* sa zostrojí analogicky ako pri $SLR(1)$, teda pri redukcii sa koná redukcia vtedy, ak je na vstupe jeden z očakávaných symbolov danej položky.
- Redukciu signalizujú položky $B \rightarrow \rho\bullet, \{I_1, \dots, I_m\}$, avšak:
 1. Pri $SLR(1)$ analyzátore $\{I_1, \dots, I_m\} = FOLLOW(B)$,
 2. Pri $LALR(1)$ analyzátore $\{I_1, \dots, I_m\} \subseteq FOLLOW(B)$.
- Preto pri $LALR(1)$ analyzátoroch vzniká menšie riziko konfliktu, a teda trieda jazykov, pre ktoré sa dá zostrojiť $LALR(1)$ analyzátor, je **väčšia**, než trieda $SLR(1)$ jazykov. Preto $SLR(1) \subset LALR(1)$.
- Ak tabuľka *ACTION* $LALR(1)$ analyzátoru neobsahuje konflikty, príslušnú gramatiku nazývame $LALR(1)$ -gramatikou.



Príklad - tabuľka *ACTION*

[illegible]

Príklad - tabuľka *GOTO*

<i>GOTO</i>	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
\$		s_6							
<i>a</i>	s_3		s_3						
<i>b</i>				s_5					
<i>c</i>	s_4		s_4						
<i>S</i>	s_8								
<i>O</i>	s_1								
<i>P</i>	s_2		s_7						



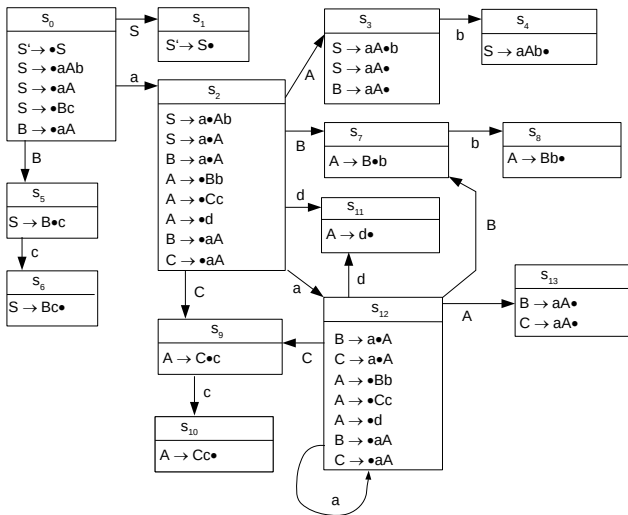
Príklad č. 2

Zostrojte $LALR(1)$ analyzátor pre gramatiku:

1. $S \rightarrow aAb$
2. $S \rightarrow aA$
3. $S \rightarrow Bc$
4. $A \rightarrow Bb$
5. $A \rightarrow Cc$
6. $A \rightarrow d$
7. $B \rightarrow aA$
8. $C \rightarrow aA$



Príklad č. 2 - $LR(0)$ automat



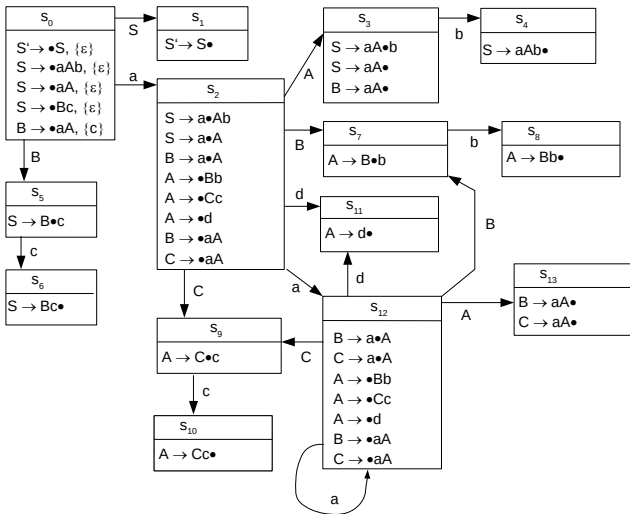
Príklad č. 2 - $LR(0)$ doplnenie očakávaných symbolov

- Konštrukcia $LALR(1)$ automatu z $LR(0)$ automatu je založená na tom, že sa teraz začnú dopĺňať k položkám očakávané symboly.
- Začnime dopĺňať očakávané symboly, najprv pre stav s_0 :

Položka	Symboly	Dôvod
$S' \rightarrow \bullet S$	ε	Počiatkové pravidlo
$S \rightarrow \bullet aAb$	ε	$CLOSURE_1(S' \rightarrow \bullet S, \{\varepsilon\})$
$S \rightarrow \bullet aA$	ε	$CLOSURE_1(S' \rightarrow \bullet S, \{\varepsilon\})$
$S \rightarrow \bullet Bc$	ε	$CLOSURE_1(S' \rightarrow \bullet S, \{\varepsilon\})$
$B \rightarrow \bullet aA$	c	$CLOSURE_1(S \rightarrow \bullet Bc, \{\varepsilon\})$



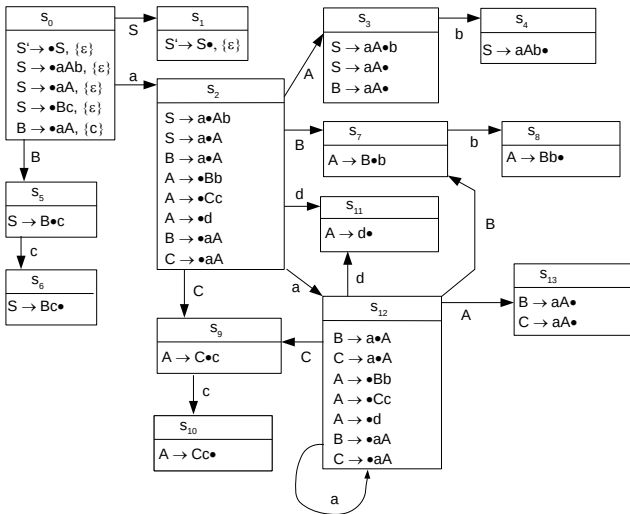
Príklad č. 2 - Doplnenie očakávaných symbolov v s_0



Doplnenie očakávaných symbolov pre stav s_1 :

Položka	Symboly	Dôvod
$S' \rightarrow S\bullet$	ε	Položka $S' \rightarrow \bullet S, \{\varepsilon\}$ zo stavu s_0

Príklad č. 2 - Doplnenie očakávaných symbolov v s_1

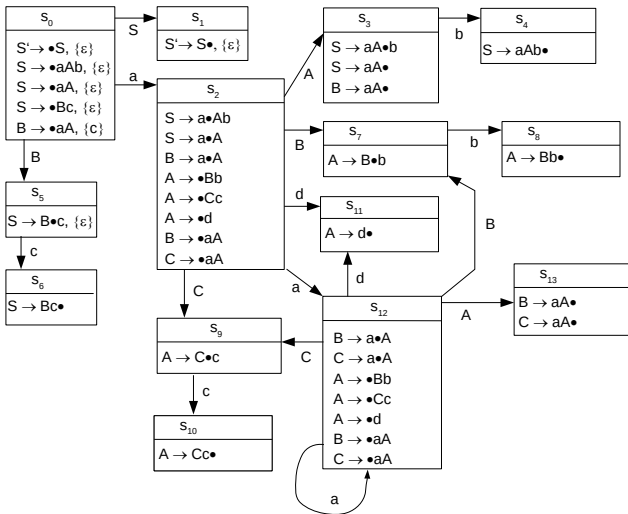


Príklad č. 2 - $LR(0)$ doplnenie očakávaných symbolov

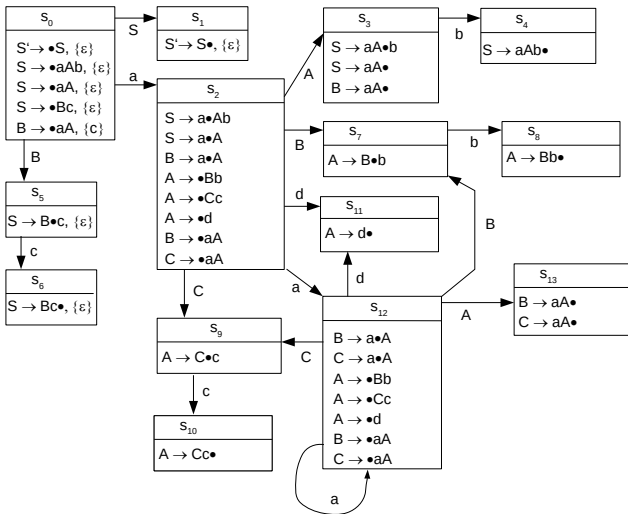
Doplnenie očakávaných symbolov pre stav s_5 :

Položka	Symboly	Dôvod
$S \rightarrow B \bullet c$	ϵ	Položka $S \rightarrow \bullet Bc$, $\{\epsilon\}$ zo stavu s_0

Príklad č. 2 - Doplnenie očakávaných symbolov v s_5



Príklad č. 2 - Doplnenie očakávaných symbolov v s_6



Pokračovanie doplnenia očakávaných symbolov pre stav s_2 :

Položka	Symboly	Dôvod
$A \rightarrow \bullet d$	ε	$CLOSURE1(S \rightarrow a \bullet A, \{\varepsilon\})$
$A \rightarrow \bullet d$	b	$CLOSURE1(S \rightarrow a \bullet Ab, \{\varepsilon\})$
$A \rightarrow \bullet d$	c	$CLOSURE1(B \rightarrow a \bullet A, \{c\})$
$B \rightarrow \bullet aA$	b	$CLOSURE1(A \rightarrow \bullet Bb, \{\varepsilon, b, c\})$
$C \rightarrow \bullet aA$	c	$CLOSURE1(A \rightarrow \bullet Cc, \{\varepsilon, b, c\})$



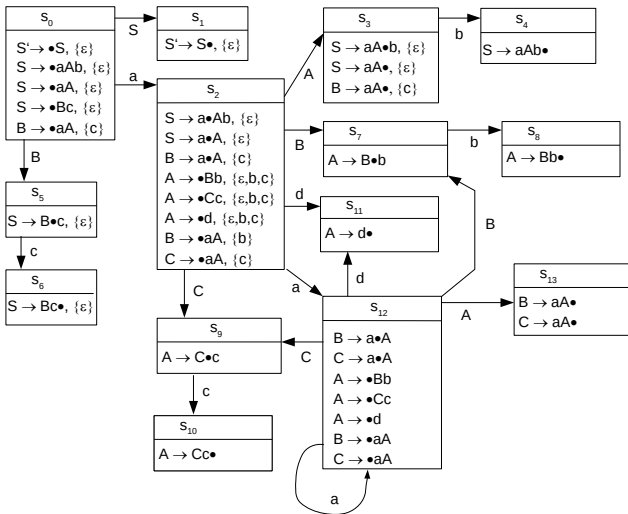
Príklad č. 2 - $LR(0)$ doplnenie očakávaných symbolov

Doplnenie očakávaných symbolov pre stav s_3 :

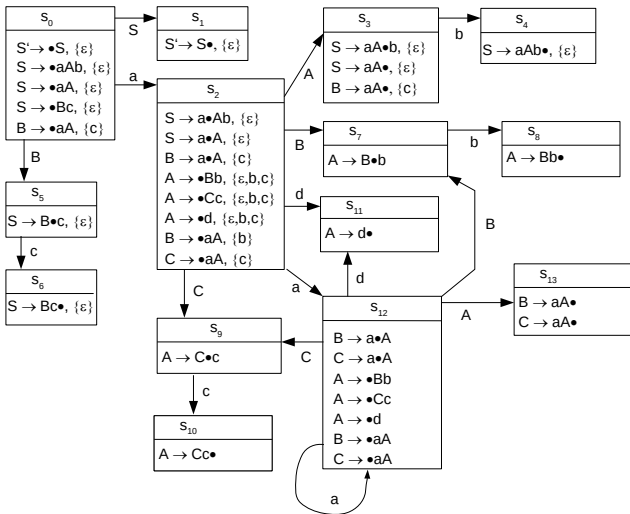
Položka	Symboly	Dôvod
$S \rightarrow aA \bullet b$	ε	Položka $S \rightarrow a \bullet Ab, \{\varepsilon\}$ zo stavu s_2
$S \rightarrow aA \bullet$	ε	Položka $S \rightarrow a \bullet A, \{\varepsilon\}$ zo stavu s_2
$B \rightarrow aA \bullet$	c	Položka $B \rightarrow a \bullet A, \{c\}$ zo stavu s_2



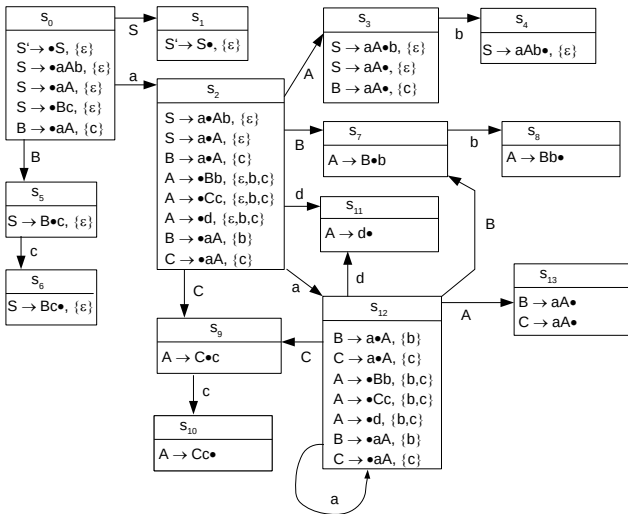
Príklad č. 2 - Doplnenie očakávaných symbolov v s_3



Príklad č. 2 - Doplnenie očakávaných symbolov v s_4



Príklad č. 2 - Doplnenie očakávaných symbolov v s_{12}



Príklad č. 2 - $LR(0)$ doplnenie očakávaných symbolov

Doplnenie očakávaných symbolov pre stav s_7 :

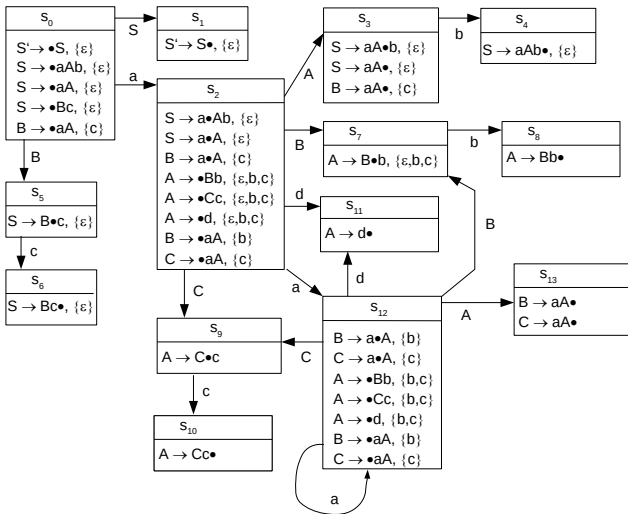
Položka	Symboly	Dôvod
$A \rightarrow B \bullet b$	ε, b, c	Položka $A \rightarrow \bullet Bb, \{\varepsilon, b, c\}$ zo stavu s_2
$A \rightarrow B \bullet b$	b, c	Položka $A \rightarrow \bullet Bb, \{b, c\}$ zo stavu s_{12}

T.j. musíme

uvažovať **všetky** relevantné stavy (položky), z ktorých idú hrany do stavu s_7 !



Príklad č. 2 - Doplnenie očakávaných symbolov v s_7

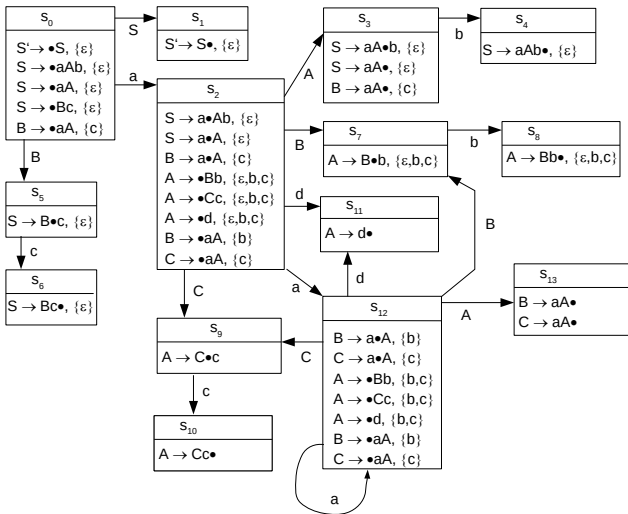


Príklad č. 2 - $LR(0)$ doplnenie očakávaných symbolov

Doplnenie očakávaných symbolov pre stav s_8 :

Položka	Symboly	Dôvod
$A \rightarrow Bb\bullet$	ε, b, c	Položka $A \rightarrow B\bullet b, \{\varepsilon, b, c\}$ zo stavu s_7

Príklad č. 2 - Doplnenie očakávaných symbolov v s_8



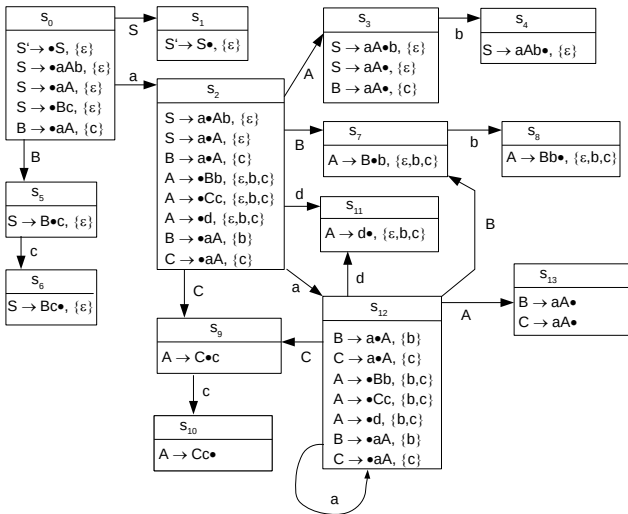
Príklad č. 2 - $LR(0)$ doplnenie očakávaných symbolov

Doplnenie očakávaných symbolov pre stav s_{11} :

Položka	Symboly	Dôvod
$A \rightarrow d\bullet$	ε, b, c	Položka $A \rightarrow \bullet d, \{\varepsilon, b, c\}$ zo stavu s_2
$A \rightarrow d\bullet$	b, c	Položka $A \rightarrow \bullet d, \{b, c\}$ zo stavu s_{12}



Príklad č. 2 - Doplnenie očakávaných symbolov v s_{11}



Príklad č. 2 - $LR(0)$ doplnenie očakávaných symbolov

Doplnenie očakávaných symbolov pre stav s_9 :

Položka	Symboly	Dôvod
$A \rightarrow C \bullet c$	ε, b, c	Položka $A \rightarrow \bullet Cc, \{\varepsilon, b, c\}$ zo stavu s_2
$A \rightarrow C \bullet c$	b, c	Položka $A \rightarrow \bullet Cc, \{b, c\}$ zo stavu s_{12}







Príklad č. 2 - ACTION&GOTO

ACTION	s0	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13
a	P		P										P	
b				P				P	R4		R5	R6		R7
c				R7		P			R4	P	R5	R6		R8
d			P										P	
e		A		R2	R1		R3		R4		R5	R6		

[illegible]

Konflikty v $LALR(1)$ -analyzátoře

- Podobne ako pri ostatných analyzátoroch, existujú jazyky / gramatiky, ktoré nie sú $LALR(1)$, t.j. pri ktorých dôjde ku vzniku konfliktov presun-redukcia alebo redukcia-redukcia.



Konflikty *LALR*(1)-analyzátoru

s_1	
...	
$E \rightarrow E + T \bullet$	$, \{id, \dots\}$
$E \rightarrow T \bullet$	$, \{id, \dots\}$
...	

Redukcia-redukcia

s_2	
...	
$T \rightarrow (E) \bullet$	$, \{id, \dots\}$
$E \rightarrow \bullet id$	$, \{\dots\}$
...	

Presun-redukcia

s_3	
...	
$T \rightarrow (E) \bullet$	$, \{+, \})\}$
$E \rightarrow (E + T) \bullet$	$, \{(\}$
$T \rightarrow (T) \bullet E$	$, \{(\text{id}\}$
$E \rightarrow \bullet id$	$, \{(\text{, } +, \text{id}\}$
...	

Bez konfliktov

Príklad č. 3

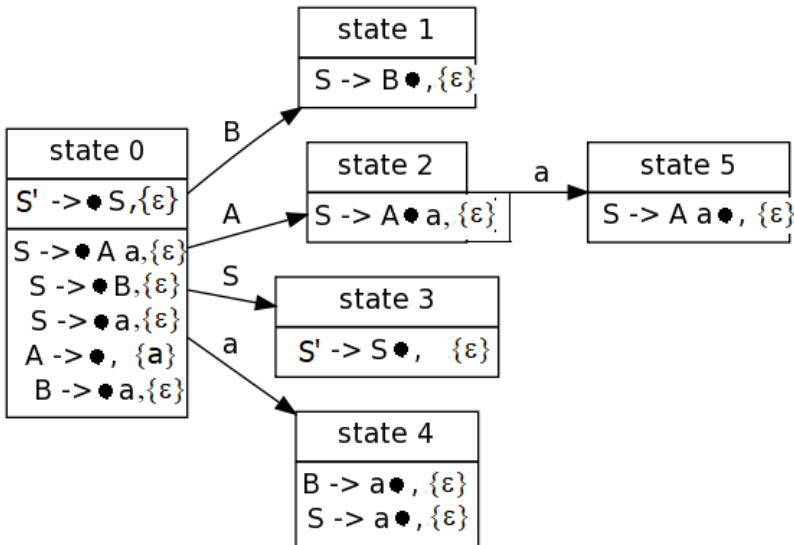
Je daná nasledovná bezkontextová gramatika $G = (N, T, P, S)$, kde

$$N = \{S, A, B\}, T = \{a\}:$$

1. $S \rightarrow Aa$
2. $S \rightarrow a$
3. $S \rightarrow B$
4. $A \rightarrow \varepsilon$
5. $B \rightarrow a$



Príklad č. 3 $LALR(1)$ -automat



Príklad č. 3 *ACTION*, *GOTO*

<i>ACTION</i>	s_0	s_1	s_2	s_3	s_4	s_5
a	P/R4		P			
ε		R3		A	R2/R5	R1

<i>GOTO</i>	s_0	s_1	s_2	s_3	s_4	s_5
<i>a</i>	s_4		s_5			
<i>S</i>	s_3					
<i>A</i>	s_2					
<i>B</i>	s_1					



Použitá literatura

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*.

