

## LR(0), SLR(k), LALR(k) a LR(k) syntaktická analýza

LR - gramatiky  $\rightarrow$  pravý pohled gramatiky

$G = (N, T, P, S)$   
 $N$  - neterminální symboly  
 $T$  - terminální symboly  
 $P$  - pravidla  
 $S$  - počáteční symbol

 $\rightarrow$  pravá derivace  $S = r_1 = r_2 = r_3 \dots = w$ - vždy se předkládá nejvíce pravý symbol  $\rightarrow$  výsledkem je pravý ~~derivace~~

pořadí pravidel pravé derivace  
 v obráceném pořadí  
 pohled

konečnouhý automata - na každou konfiguraci je představena pravá větná forma  
 $\rightarrow$  spolu s nepřevodem vstupem celá pravá větná forma

$$r_i = \alpha A a_j a_{j+1} \dots a_n : A \in N, \alpha \in (NUT)^*, a_j a_{j+1} \dots a_n \in T^*$$

transitions:  $\delta(q, a, \varepsilon) = \{(q, a) \mid \forall a \in T\}$  - shift $\delta(q, \varepsilon, \alpha) = \{(q, A) \mid A \rightarrow \alpha \in P\}$  - reduce $\delta(q, \varepsilon, \#S) = \{(r, \varepsilon)\}$  - accept

star automata  $\uparrow$   
 vstupní symbol  $\uparrow$   
 obsah každé konfigurace  $\uparrow$   
 nový star automata  $\uparrow$   
 nový obsah konfigurace

minimální: FIRST - pro každé pravidlo  $\rightarrow$  jaký první symbol může být po jeho nahrazení

$$S \rightarrow \varepsilon \rightarrow \varepsilon$$

$$S \rightarrow aA \rightarrow a$$

$$S \rightarrow Ab \rightarrow \text{FIRST}(A) \text{ \& } b \text{ pokud } \text{FIRST}(A) \text{ obsahuje } \varepsilon$$

FOLLOW - pro každý neterminální symbol - co se může objevit na něm

$$S \rightarrow Ab \rightarrow A: b$$

$$A \rightarrow B \rightarrow \text{B-FOLLOW}(A)$$

$$S \rightarrow X \rightarrow X: \varepsilon$$

$$\text{FOLLOW}(A) = S \rightarrow^* \alpha A \beta, \alpha \in \text{FIRST}(\beta)$$

# LR(0) - parser

- nejjednodušší parser, který nerozhoduje na základě symbolů na vstupu (dopředu prohlížených)
- rozhoduje se pouze na základě kontextu  $\leftarrow$  strong LR - nepotřebuje historii, pouze všech kontextu  
weak LR - používá i historii
- parser používá k rozhodování LR automata (charakteristický automata)
  - každý stav reprezentuje jeden stav kontextu
  - ve větách (listech) jsou redukce - jinak konflikt

## • sestavení parseru: LR(0) položky

→ začínáme s rozšířenou bezkontextovou grammatikou

$$G' = (N \cup \{S'\}, T, P \cup \{S' \rightarrow S\}, S')$$

→ sestavíme množiny položek

$$\# := \{S' \rightarrow \cdot S\}$$

- každá množina (= stav charakteristického automatu) se jmenuje dle symbolu před tečkou

- třeba knač, kde se momentálně nacházíme ve čtení

- uzavřené množiny: pokud je před N symbolem, expandují (přidávají) i všechna pravidla začínající N

→ opakují dokud nemáme všechny

příklad: 1:  $S' \rightarrow S$

2:  $S \rightarrow aA$

3:  $A \rightarrow bbBb$

4:  $B \rightarrow c \mid d$

$$\# = \{S' \rightarrow \cdot S, S \rightarrow \cdot aA\}$$

$$a_1 = \{S \rightarrow a \cdot A, A \rightarrow \cdot bbBb\}$$

$$B = \{A \rightarrow bbB \cdot b\}$$

$$b_3 = \{A \rightarrow bbBb \cdot\}$$

$$S = \{S' \rightarrow S \cdot\}$$

$$A = \{S \rightarrow aA \cdot\}$$

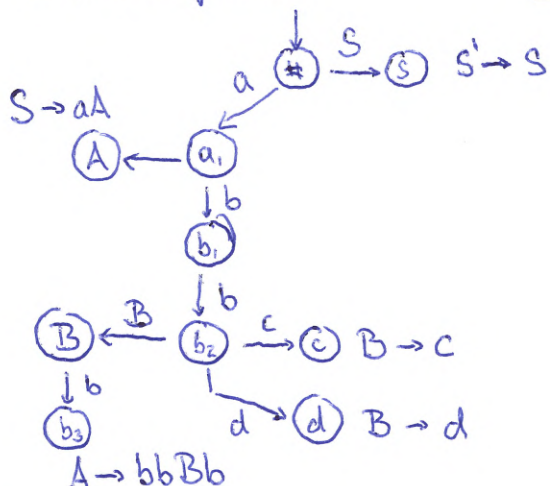
$$b_1 = \{A \rightarrow b \cdot bBb\}$$

$$b_2 = \{A \rightarrow bb \cdot Bb, B \rightarrow \cdot c, B \rightarrow \cdot d\}$$

$$c = \{B \rightarrow c \cdot\}$$

$$d = \{B \rightarrow d \cdot\}$$

charakteristický automata:



- gramatika není řiditelná - existují více b množin

- při čtení na kontextu ukládám symboly dle pojmenování skupin!

- redukci  $S' \rightarrow S$  končíme úspěchem (pokud není vstup)



výpočtová pomocná tabuľka: (rozkladová)

MI-SP-SP-20

2

- na každom riadku môžu byť všetky symboly ze jakej skupiny

#	P	
S	Accept	- is koniec stavu je redukcia pravidlom $S \rightarrow S$ , čo je ekvivalentný prechod
a <sub>1</sub>	P	
A	R(2)	príklad: Input = abbdb
b <sub>1</sub>	P	$(\#, abbdb, \epsilon) \vdash (\#a, bddb) \vdash (\#a, b_1, bddb, \epsilon) \vdash$
b <sub>2</sub>	P	$\vdash (\#a, b_1, b_2, db, \epsilon) \vdash (\#a, b_1, b_2, d, b, \epsilon) \vdash (\#a, b_1, b_2, B, b, 5) \vdash$
B	P	
b <sub>3</sub>	R(3)	$\vdash (\#a, b_1, b_2, B, b_3, \epsilon, 5) \vdash (\#a, A, \epsilon, 53) \vdash (\#S, \epsilon, 532) \vdash$
c	R(4)	
d	R(5)	$\vdash (\text{accept}) \checkmark$

## SLR(k) - parser

- občas máme konflikt shift reduce, ktorý LR(0) nemôže riešiť

→ rozhodujú na rozklade FIRST & FOLLOW množín pre dané symboly (stav)

- množiny se vytvárajú z pôvodných pravidiel = väčšie množiny konštrukcií

- 1  $E' \rightarrow E$
- 2  $E \rightarrow E + T$
- 3  $E \rightarrow T$
- 4  $T \rightarrow T * F$
- 5  $T \rightarrow F$
- 6  $F \rightarrow (E)$
- 7  $F \rightarrow \epsilon$

- LR(0) problémy obsluhuje konflikty:

$$E_1 = \{E' \rightarrow E., E \rightarrow E. + T\}$$

FOLLOW

$$\text{FOLLOW}(E') = \{\epsilon\} \quad \text{FIRST}(+T) = \{+\}$$

→ máme disjunktné množiny, môžeme se deterministicky rozhodnúť

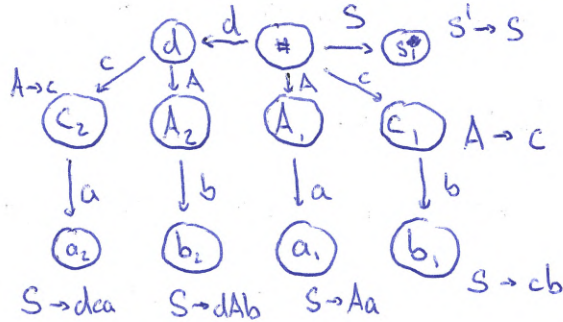
- rozkladová tabuľka má stĺpce pre každý T symbol

- vyplníme redukcie a symboly ze FOLLOW daného stavu (by co majú v grafe prechod)  
+ prázdny prechod

F	a	+	*	(	)	$\epsilon$
#	P			P		
E <sub>1</sub>		P				accept
T <sub>1</sub>		R(2)	P		R(2)	R(2)

LALR (k) - parser

- kein' problemu is medijunklunimi FIRST & FOLLOW
- is polskate is chitunigi per kachy' shor kachid' (FOLLOW)

$$0: S' \rightarrow S \quad 3: S \rightarrow cb$$
$$1: S \rightarrow Aa \quad 4: S \rightarrow dca$$
$$2: S \rightarrow dAb \quad 5: A \rightarrow a$$


FOLLOW = {a, b}

FIRST = {b}

$\Rightarrow$  konflikt

- pozitivne body LR(k) položky

$[A \rightarrow \alpha\beta, w]$   $w \in T^*$  - dopředná prohlášený řešení

$$\Rightarrow C_1 = \{[S \rightarrow c.b, \epsilon], [A \rightarrow c., a]\} \quad C_2 = \{[S \rightarrow d.e.a, \epsilon], [A \rightarrow c., b]\}$$

- dopředu prohlášení některé moci před vykoukal z autorky
- vyhledem je další symbol ke expandování nekonečným

LR(k) - parser

- $\text{LR}(k)$  kontlikty redukcije - redukcije (napišit ne vimech?)
- LALR ~~stoj~~  $\text{LR}(k)$  množiny ne množinám jadrům (a napiš vimech)
- $\rightarrow \text{LR}(k)$  je pojmal nebudou

- veliki manuskripta veličnost - až  $2^{161}$  stranica