

Parsiranje od dna ka vrhu

(Nastavak)

Konstrukcija SR parsera za SLR(1) gramatike

- SLR(1) = Simple LR(1)
- ideja je da parser gleda tekući ulazni simbol pri radu
- time se smanjuje broj potencijalnih konflikata u kontrolnoj tabeli
- REDUCE($\langle X \rangle \rightarrow \gamma$) biće izvršeno samo ako je tekući ulaz iz skupa FOLLOW($\langle X \rangle$)

Follow skupovi neterminala

- FOLLOW skup posmatranog neterminala $\langle X \rangle$ je skup terminala koji se mogu pojaviti neposredno iza $\langle X \rangle$ u bilo kojoj sentencijalnoj formi izvedenoj iz $\langle S \rangle \mid$ (marker kraja je dodat jer se svaki ulazni niz njime završava).

$$\text{FOLLOW}(\langle X \rangle) = \{t \mid t \in V_t \cup \{\mid\} \\ \wedge \langle S \rangle \mid \xRightarrow{*} \alpha \langle X \rangle t \gamma\}$$

Računanje FOLLOW skupova

- Pojmovi na kojima se zasniva računanje FOLLOW skupova su poništivost neterminala i FIRST skupovi neterminala.

Poništivost neterminala

- Neterminal $\langle X \rangle$ je poništiv ako i samo ako se iz njega može izvesti prazna sekvenca:
$$\langle X \rangle \xRightarrow{*} \varepsilon$$
- Sekvenca gramatičkih simbola je poništiva ako i samo ako se sastoji od nula ili više poništivih neterminala.
- Skup poništivih neterminala P formiramo na sledeći način:
 1. U skup P staviti sve neterminale koji se pojavljuju na levoj strani neke prazne smene.
 2. Razmatrati redom smene čije se desne strane sastoje isključivo od neterminala, a leva strana se ne nalazi u skupu P . Ako se svi neterminali desne strane pojavljuju u skupu P , dodati levu stranu smene u P .
 3. Sve dok postoji promena u skupu P ponavljati korak 2, u suprotnom završiti postupak.

First skupovi

- Skup $\text{FIRST}(<X>)$ za proizvoljan neterminal $<X>$ predstavlja skup svih terminalnih simbola kojima može početi sentencijalna forma izvedena iz $<X>$:
$$\text{FIRST}(<X>) = \{ t \mid t \in V_t \wedge \alpha \in V^* \wedge <X> \Rightarrow^* t \alpha \}$$
- Moguće je definisati FIRST skupove sentencijalnih formi: Neka je $\alpha = A_1 A_2 \dots A_i A_{i+1} \dots A_n$ proizvoljna sentencijalna forma u kojoj prvih i simbola predstavlja poništive neterminale. Tada skup $\text{FIRST}(\alpha)$ predstavlja uniju FIRST skupova svih simbola sa levog kraja sekvence zaključno sa prvim koji nije poništiv:
$$\text{FIRST}(\alpha) = \text{FIRST}(A_1) \cup \text{FIRST}(A_2) \cup \dots \cup \text{FIRST}(A_{i+1})$$
- Ukoliko je simbol A terminal, onda je $\text{FIRST}(A) = \{A\}$.

Iterativno računanje FIRST skupova neterminala:

1. Inicijalno, FIRST skupovi svih neterminala su prazni.
2. Razmatramo redom svaki neterminal $<X>$
3. Razmatramo redom sve smene sa levom stranom $<X>$.
4. Neka je jedna takva smena $<X> \rightarrow \alpha$. Tada računamo:
 - a. $\text{FIRST}(<X>) = \text{FIRST}(<X>) \cup \text{FIRST}(\alpha)$
5. Tačke 2-4 algoritma ponavljamo sve dok ima promene u FIRST skupovima.

Računanje FOLLOW skupova

- FOLLOW skupove računamo prema sledećim pravilima:

1. Skup FOLLOW startnog neterminala sadrži marker kraja ulaza $\mid$.
2. Za određivanje FOLLOW skupa neterminala $\langle X \rangle$ posmatramo sve gramatičke smene na čijoj se desnoj strani nalazi $\langle X \rangle$. Neka je $\langle Y \rangle \rightarrow \alpha \langle X \rangle \beta$ jedna od tih smena. Tada:

$\text{FOLLOW}(\langle X \rangle) \supset \text{FIRST}(\beta)$, ako β nije poništivo, ili
 $\text{FOLLOW}(\langle X \rangle) \supset \text{FIRST}(\beta) \cup \text{FOLLOW}(\langle Y \rangle)$, ako je β poništivo.

- U opštem slučaju, na osnovu datih pravila formira se sistem skupovnih jednakosti koji se rešava iterativno:

1. Inicijalno se usvoji da su svi FOLLOW skupovi prazni.
2. Usvojene vrednosti se uvrste u jednakosti čime se dobijaju nove vrednosti FOLLOW skupova.
3. Korak 2. ponavlja se sve dok postoji promena bar u jednom skupu. Krajnje vrednosti predstavljaju konačno rešenje.

Primer računanja FOLLOW skupova:

Za sledeću gramatiku sa startnim simbolom $\langle A \rangle$:

1. $\langle A \rangle \rightarrow a \langle B \rangle \langle C \rangle$

2. $\langle A \rangle \rightarrow b \langle B \rangle$

3. $\langle A \rangle \rightarrow \varepsilon$

4. $\langle B \rangle \rightarrow a \langle B \rangle b$

5. $\langle B \rangle \rightarrow \varepsilon$

6. $\langle C \rangle \rightarrow b \langle C \rangle$

7. $\langle C \rangle \rightarrow c$

Odrediti FOLLOW skupove za svaki neterminal.

Rešenje

Odredimo prvo poništive neterminale u datoj gramatici, zatim FIRST, a potom i FOLLOW skupove.

1. $\langle A \rangle \rightarrow a \langle B \rangle \langle C \rangle$	$Pon = \{ \langle A \rangle, \langle B \rangle \}$
2. $\langle A \rangle \rightarrow b \langle B \rangle$	$FIRST(\langle A \rangle) = \{a, b\}$
3. $\langle A \rangle \rightarrow \varepsilon$	$FIRST(\langle B \rangle) = \{a\}$
4. $\langle B \rangle \rightarrow a \langle B \rangle b$	$FIRST(\langle C \rangle) = \{b, c\}$
5. $\langle B \rangle \rightarrow \varepsilon$	$FOLLOW(\langle A \rangle) = \{ \text{—} \}$
6. $\langle C \rangle \rightarrow b \langle C \rangle$	$FOLLOW(\langle B \rangle) \supset FIRST(\langle C \rangle)$
7. $\langle C \rangle \rightarrow c$	$FOLLOW(\langle B \rangle) \supset FIRST(\varepsilon) \cup FOLLOW(\langle A \rangle)$
	$FOLLOW(\langle B \rangle) \supset \{b\}$
	$FOLLOW(\langle B \rangle) = FIRST(\langle C \rangle) \cup FOLLOW(\langle A \rangle) \cup \{b\}$
	$= \{b, c, \text{—} \}$
	$FOLLOW(\langle C \rangle) \supset FIRST(\varepsilon) \cup FOLLOW(\langle C \rangle)$
	$FOLLOW(\langle C \rangle) \supset FIRST(\varepsilon) \cup FOLLOW(\langle A \rangle) = \{ \text{—} \}$

Konstrukcija SLR(1) parsera

- Karakteristični automat i potisna tabela SLR(1) parsera identični su karakterističnom automatu i potisnoj tabeli LR(0) parsera.

Popunjavanje kontrolne SLR(1) tabele

- (razmatra se ulaz u vrsti V , kojoj odgovara skup konfiguracija S i koloni x):

1. Ako S sadrži konfiguraciju oblika $\langle X \rangle \rightarrow \alpha \bullet x \beta$, odnosno tačka se nalazi ispred terminalnog simbola x , ulaz treba popuniti akcijom **SHIFT**.
2. Ako S sadrži konfiguraciju oblika $\langle Y \rangle \rightarrow \gamma \bullet$, odnosno tačka se nalazi na kraju desne strane, a važi da je $x \in \text{FOLLOW}(\langle Y \rangle)$, tada ulaz treba popuniti akcijom **REDUCE**($\langle Y \rangle \rightarrow \gamma$). Izuzetno, ako se radi o nultoj smeni, akcija je **ACCEPT**.
3. Vrste koje nisu pokrivene ni 1. ni 2. pravilom treba da sadrže akciju **REJECT**.

Ukoliko gornja pravila na jednoznačan način određuju akciju za svaki ulaz kontrolne tabele, gramatika pripada klasi SLR(1).

SLR(1) - Primer za ranije datu gramatiku

0. $\langle S' \rangle \rightarrow \langle S \rangle \text{---|}$

1. $\langle S \rangle \rightarrow a$

2. $\langle S \rangle \rightarrow (\langle S \rangle \langle R \rangle$

3. $\langle R \rangle \rightarrow , \langle S \rangle \langle R \rangle$

4. $\langle R \rangle \rightarrow)$

$\text{FOLLOW}(\langle S' \rangle) = \{ \text{---|} \}$

$\text{FOLLOW}(\langle S \rangle) = \{ (,) \text{---|} \}$

$\text{FOLLOW}(\langle R \rangle) = \{ ,) \text{---|} \}$

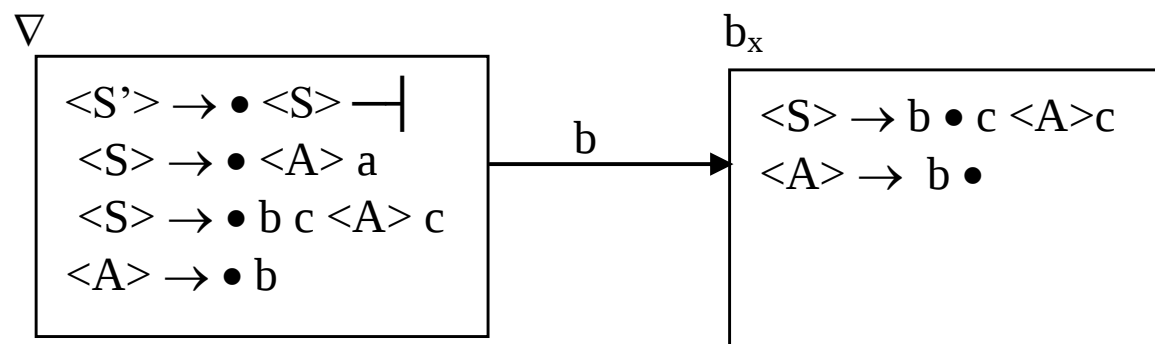
	(	,	)	a	---
∇	SHIFT			SHIFT	
$\langle S \rangle_0$					SHIFT
--- _0					ACCEPT
a_1		REDUCE(1)	REDUCE(1)		REDUCE(1)
$(_2$	SHIFT			SHIFT	
$\langle S \rangle_2$		SHIFT	SHIFT		
$\langle R \rangle_2$		REDUCE(2)	REDUCE(2)		REDUCE(2)
$,_3$	SHIFT			SHIFT	
$\langle S \rangle_3$		SHIFT	SHIFT		
$\langle R \rangle_3$		REDUCE(3)	REDUCE(3)		REDUCE(3)
$)_4$		REDUCE(4)	REDUCE(4)		REDUCE(4)

LALR(1) parsiranje

Nedostatak SLR(1) metoda

- Gramatika

0. $\langle S' \rangle \rightarrow \langle S \rangle \mid$
1. $\langle S \rangle \rightarrow \langle A \rangle a$
2. $\langle S \rangle \rightarrow b c \langle A \rangle c$
3. $\langle A \rangle \rightarrow b$



nije SLR(1) jer se u kontrolnoj tabeli pojavljuje S/R konflikt u stanju b_x i koloni c jer je $\text{FOLLOW}(\langle A \rangle) = \{a, c\}$:

- konflikta ne bi bilo kada bi se uočilo da $\text{REDUCE}(\langle A \rangle \rightarrow b)$ ne treba raditi u koloni c , pošto $\langle A \rangle$ „dolazi” iz smene 2, tako da R treba raditi samo u koloni a , a SHIFT u koloni c
- problem je što SLR(1) parser pri redukciji koristi FOLLOW skup, koji uzima u obzir i $\langle A \rangle$ u drugoj smeni (tj. u FOLLOWu se računaju sve pojave odjednom)
- dakle, neki konflikti mogu se izbeći ako se uzmu precizniji skupovi od FOLLOW pri popunjavanju kontrolne tabele
- ovo je ostvareno kod LALR(1) gramatika

Pojam LR(1) konfiguracije

- **LR(1) konfiguracija** predstavlja LR(0) konfiguraciju sa dodatim **predikcionim simbolom**

$$A \rightarrow X_1 \dots X_i \bullet X_{i+1} \dots X_n, p \quad p \in V_T \cup \{\varepsilon\}$$

(ε se koristi kao predikcija samo kod konfiguracija smene 0.)

Primer

$$0. \langle S' \rangle \rightarrow \langle S \rangle - |$$

$$1. \langle S \rangle \rightarrow \langle A \rangle \langle A \rangle \langle S \rangle$$

$$2. \langle S \rangle \rightarrow \varepsilon$$

$$3. \langle S \rangle \rightarrow a$$

$$4. \langle A \rangle \rightarrow a$$

$$\langle A \rangle \rightarrow *a, -|$$

Neke LR(1) konfiguracije:

$$\langle S' \rangle \rightarrow \bullet \langle S \rangle -|, \varepsilon$$

$$\langle S \rangle \rightarrow \bullet a, -|$$

$$\langle A \rangle \rightarrow a \bullet, a$$

$$\langle A \rangle \rightarrow *a, a \quad \langle A \rangle \rightarrow *a, \{a, -|\}$$

Pravilne LR(1) konfiguracije

- u okviru simbola steka LALR(1) parsera može se pojaviti samo podskup LR(1) konfiguracija koje nazivamo *pravilnim*.
- LR(1) konfiguraciju $\langle X \rangle \rightarrow \alpha \bullet \beta$, t nazivamo **pravilnom LR(1) konfiguracijom** ako i samo ako je:
 1. $t \in \text{FOLLOW}(\langle X \rangle)$, ili
 2. reč o smeni 0 i $t = \epsilon$.

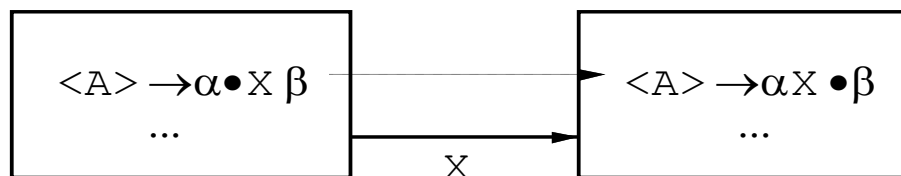
Iterativni metod konstrukcije LALR(1) parsera

- 1) konstruisanje karakterističnog LR(0) automata i
- 2) određivanje predikcionih skupova (engl. lookahead sets) LA **metodom prosleđivanja predikcionih simbola** (engl. *lookahead propagation*).

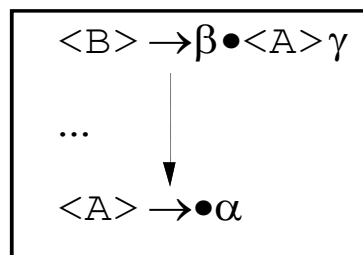
Metod prosleđivanja predikcionih simbola:

- 1) Odrediti skupove **spontanih predikcionih simbola** za sve konfiguracije zatvaranja u svakom od stanja LR(0) detektora ručki:
 - SLA skup za proizvoljnu LR(0) konfiguraciju zatvaranja $\langle A \rangle \rightarrow \bullet \alpha$ u stanju S računa se tako što se:
 - a) nađu sve konfiguracije oblika $\langle B \rangle \rightarrow \beta \bullet \langle A \rangle \gamma$ u stanju S i
 - b) u skup $SLA(\langle A \rangle \rightarrow \bullet \alpha)$ uključe svi terminali $t \in FIRST(\gamma)$.
- 2) Između LR(0) konfiguracija u stanjima LR(0) automata definišu se veze *prosleđivanja* prema slici:

1. slučaj



2. slučaj



γ je poništivo



prelaz između stanja



veza prosleđivanja

Predikcioni (lookahead) skupovi

LA skupovi računaju se sledećim iterativnim algoritmom:

- Inicijalno su LA skupovi konfiguracija zatvaranja u svakom od stanja jednaki SLA skupovima odgovarajućih konfiguracija, dok su LA skupovi ostalih konfiguracija prazni.
- Predikcioni simboli prosleđuju se među LA skupovima konfiguracija koje su povezane vezama prosleđivanja sve dok postoji promena u LA skupovima.

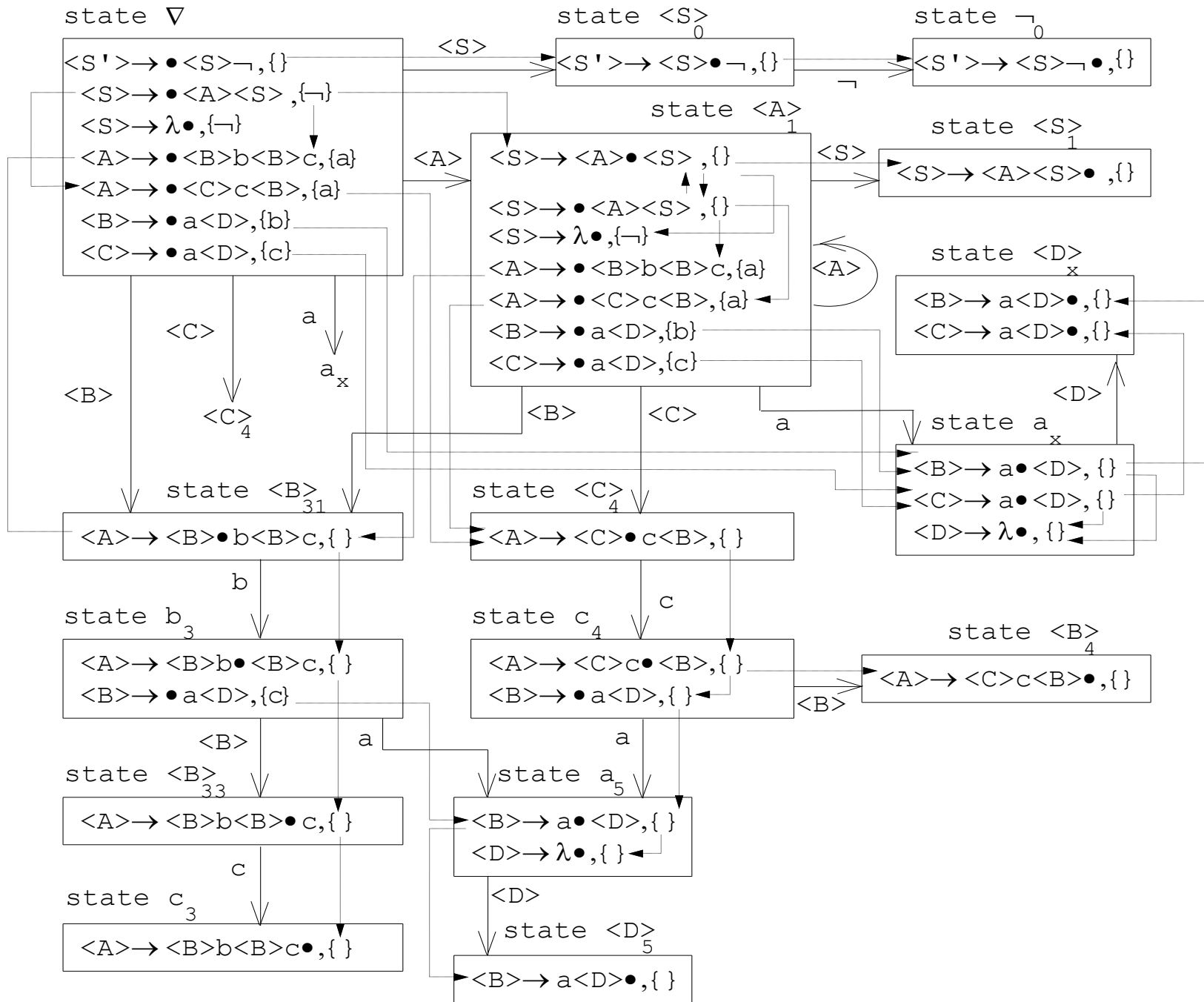
Primer

Za gramatiku:

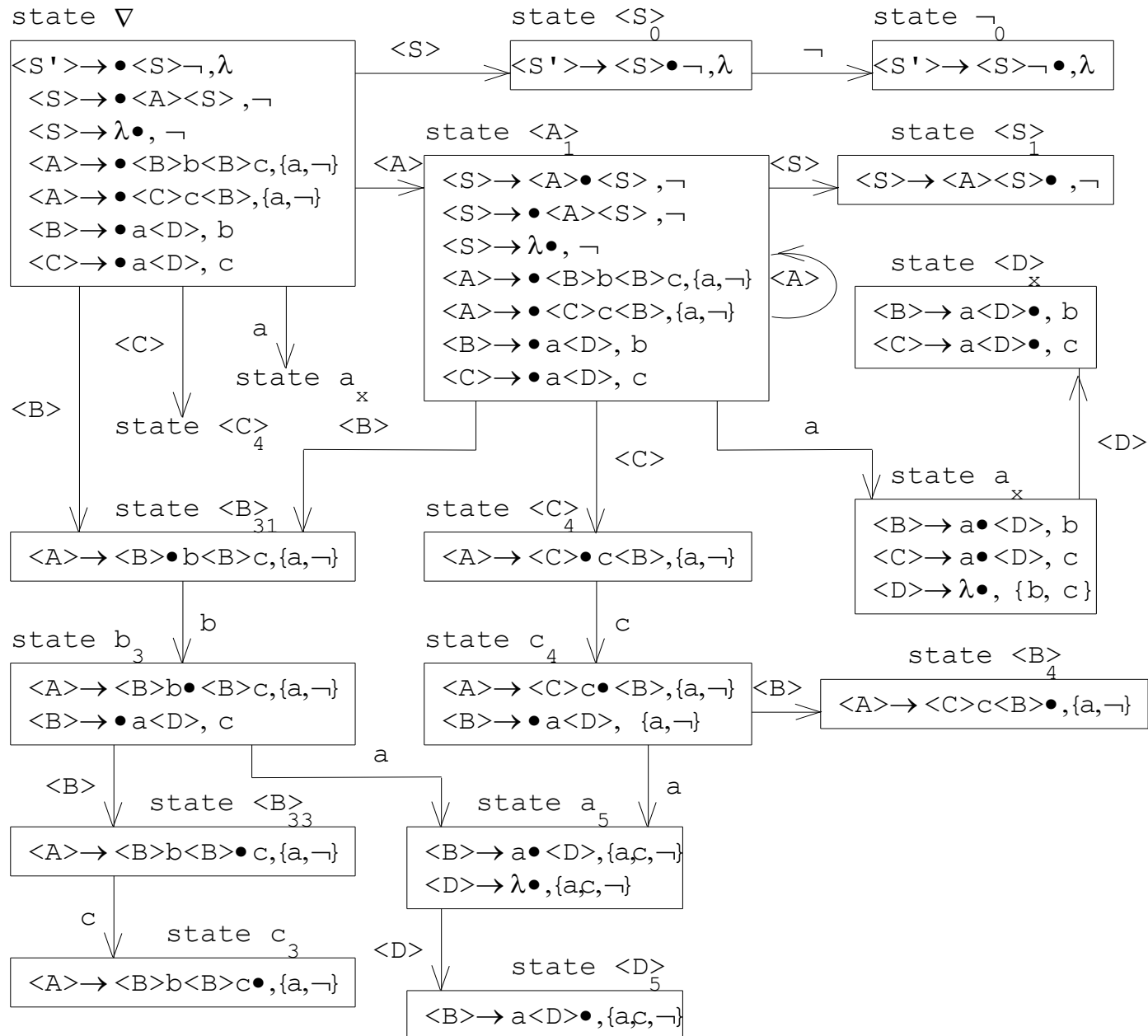
1. $\langle S \rangle \rightarrow \langle A \rangle \langle S \rangle$
2. $\langle S \rangle \rightarrow \lambda$
3. $\langle A \rangle \rightarrow \langle B \rangle b \langle B \rangle c$
4. $\langle A \rangle \rightarrow \langle C \rangle c \langle B \rangle$
5. $\langle B \rangle \rightarrow a \langle D \rangle$
6. $\langle C \rangle \rightarrow a \langle D \rangle$
7. $\langle D \rangle \rightarrow \lambda$

Sledeća slika prikazuje stanje početno stanje pri računanju predikcionih simbola za ovu gramatiku.

- LR(0) konfiguracijama u svakom stanju LR(0) detektora ručki dodati su SLA skupovi predikcija.
- Isprekidane linije predstavljaju veze prosleđivanja.



- Računanjem LA skupova određen je LALR(1) detektor ručki (ili karakteristični automat). Karakteristični automat ujedno predstavlja potisnu tabelu.



Konstrukcija kontrolne LALR(1) tabele

(razmatra se ulaz u vrsti X i koloni y):

- neprazna potisna tabela za u vrsti X i koloni y => **SHIFT**
- Ako $\langle W \rangle \rightarrow \alpha \bullet$, $y \in X$ => **REDUCE**($\langle W \rangle \rightarrow \alpha$)
- REDUCE nulte smene => **ACCEPT**
- Svi ulazi tabele koji nisu obuhvaćeni gornjim pravilima sadrže **REJECT** akciju.
- Ako svaki ulaz u kontrolnoj tabeli, posle njihovog popunjavanja prema navedenim pravilima, sadrži jedinstvenu akciju zadata gramatika pripada klasi LALR(1) gramatika i konstrukcija parsera je okončana.
- U suprotnom slučaju, zadata gramatika ne pripada klasi LALR(1) gramatika, pa se za nju ne može konstruisati LALR(1) parser.

Odnos među LR klasama bezkontekstnih gramatika

- od svih LR metoda LALR(1) parsiranje ima za praksu najveći značaj.
- Parseri generisani alatima yacc i cup su LALR(1) parseri

