

Introduction

Construction of The Canonical
LR(1) Collection

Construction of The Canonical LR(1) Collection

- ***Algorithm:***

C is { closure($\{S' \rightarrow .S, \$\}$) }

repeat the followings until no more set of LR(1) items can be added to **C** .

for each I in **C** and each grammar symbol X

if $\text{goto}(I, X)$ is not empty and not in **C**

 add $\text{goto}(I, X)$ to **C**

- goto function is a DFA on the sets in C .

A Short Notation for The Sets of LR(1) Items

- A set of LR(1) items containing the following items

$$A \rightarrow \alpha \cdot \beta, a_1$$

...

$$A \rightarrow \alpha \cdot \beta, a_n$$

can be written as

$$A \rightarrow \alpha \cdot \beta, \{a_1, a_2, \dots, a_n\}$$

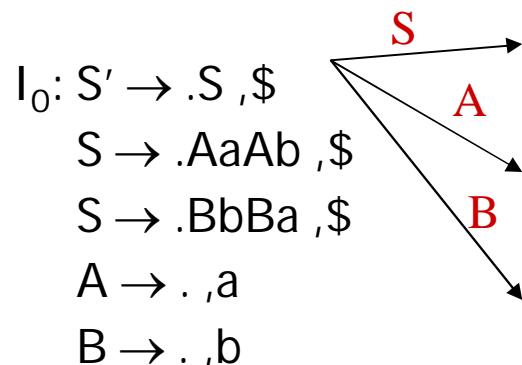
Canonical LR(1) Collection -- Example

$$S \rightarrow AaAb$$

$$S \rightarrow BbBa$$

$$A \rightarrow \epsilon$$

$$B \rightarrow \epsilon$$



$$I_1: S' \rightarrow S. , \$$$

\xrightarrow{a} to I_4

$$I_2: S \rightarrow A.aAb , \$$$

\xrightarrow{b} to I_5

$$I_3: S \rightarrow B.bBa , \$$$

$$I_4: S \rightarrow Aa.Ab , \$$$

\xrightarrow{A}

$A \rightarrow ., b$

$$I_6: S \rightarrow AaA.b , \$$$

\xrightarrow{a}

$$I_8: S \rightarrow AaAb. , \$$$

$$I_5: S \rightarrow Bb.Ba , \$$$

\xrightarrow{B}

$B \rightarrow ., a$

$$I_7: S \rightarrow BbB.a , \$$$

\xrightarrow{b}

$$I_9: S \rightarrow BbBa. , \$$$

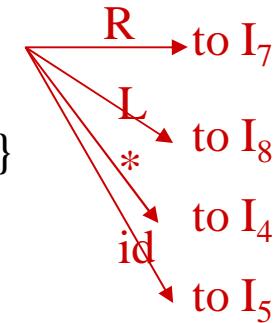
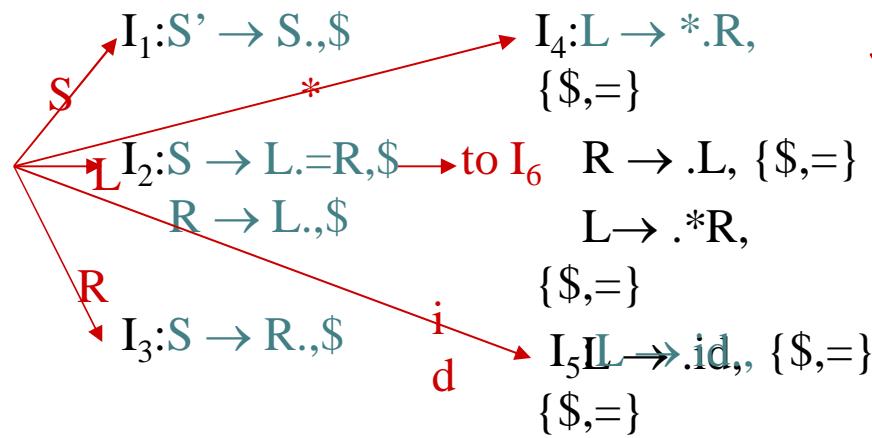
Canonical LR(1) Collection - Example2

$S' \rightarrow S$ $I_0: S' \rightarrow .S, \$$
 1) $S \rightarrow L=R$ $S \rightarrow .L=R, \$$
 2) $S \rightarrow R$ $S \rightarrow .R, \$$
 3) $L \rightarrow *R$ $L \rightarrow$
 4) $L \rightarrow id$ $.*R, \{\$, =\}$
 5) $R \rightarrow L$ $L \rightarrow .id,$
 $\{\$, =\}$

$R \rightarrow .L, \$$
 $I_6: S \rightarrow L=R, \$$
 $R \rightarrow .L, \$$
 $L \rightarrow .*R, \$$
 $L \rightarrow .id, \$$

$I_7: L \rightarrow *R., \{\$, =\}$

$I_8: R \rightarrow L., \{\$, =\}$



$I_9: S \rightarrow L=R., \$$

$I_{10}: R \rightarrow L., \$$

$I_{11}: L \rightarrow *.R, \$$
 $R \rightarrow .L, \$$
 $L \rightarrow .*R, \$$
 $L \rightarrow .id, \$$

$I_{12}: L \rightarrow id., \$$

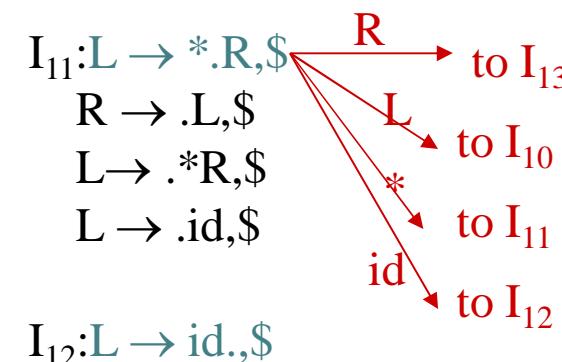
$I_{13}: L \rightarrow *.R., \$$

I_4 and I_{11}

I_5 and I_{12}

I_7 and I_{13}

I_8 and I_{10}



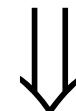
Construction of LR(1) Parsing Tables

1. Construct the canonical collection of sets of LR(1) items for G' .
 $C \leftarrow \{I_0, \dots, I_n\}$
2. Create the parsing action table as follows
 - If a is a terminal, $A \rightarrow \alpha . a \beta, b$ in I_i and $\text{goto}(I_i, a) = I_j$ then $\text{action}[i, a]$ is ***shift j***.
 - If $A \rightarrow \alpha . , a$ is in I_i , then $\text{action}[i, a]$ is ***reduce A → α*** where $A \neq S'$.
 - If $S' \rightarrow S . , \$$ is in I_i , then $\text{action}[i, \$]$ is ***accept***.
 - If any conflicting actions generated by these rules, the grammar is not LR(1).
3. Create the parsing goto table
 - for all non-terminals A , if $\text{goto}(I_i, A) = I_j$ then $\text{goto}[i, A] = j$
4. All entries not defined by (2) and (3) are errors.
5. Initial state of the parser contains $S' \rightarrow .S, \$$

LR(1) Parsing Tables - (for Example2)

	id	*	=	\$	S	L	R
0	s5	s4			1	2	3
1				acc			
2			s6	r5			
3				r2			
4	s5	s4				8	7
5			r4	r4			
6	s12	s11				10	9
7			r3	r3			
8			r5	r5			
9				r1			
10				r5			
11	s12	s11				10	13
12				r4			
13				r3			

no shift/reduce or
no reduce/reduce conflict



so, it is a LR(1) grammar