Introduction

Construction of The Canonical LR(1) Collection

Construction of The Canonical LR(1) Collection

- ***Algorithm***:

  C is { closure({S'→.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 goto(I,X) is not empty and not in ***C***

  add 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

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

...

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

can be written as

$$A \rightarrow \alpha \bullet \beta, \{a_1, a_2, ..., a_n\}$$

Canonical LR(1) Collection -- Example

$S \to AaAb$
$S \to BbBa$
$A \to \varepsilon$
$B \to \varepsilon$

I_0: $S' \to .S\ ,\$$
$S \to .AaAb\ ,\$$
$S \to .BbBa\ ,\$$
$A \to .\ ,a$
$B \to .\ ,b$

S, A, B

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

a → to I_4

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

b → to I_5

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

I_4: $S \to Aa.Ab\ ,\$$
$A \to .\ ,b$

A →

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

a →

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

I_5: $S \to Bb.Ba\ ,\$$
$B \to .\ ,a$

B →

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

b →

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

Canonical LR(1) Collection – Example2

S' → S
1) S → L=R
2) S → R
3) L→ *R
4) L → id
5) R → L

I_0: S' → .S,$
S → .L=R,$
S → .R,$
L → .*R,{$,=}
L → .id, {$,=}
R → .L,$

S: I_1: S' → S.,$
*: I_4: L → *.R, {$,=}
R → .L, {$,=}
L→ .*R, {$,=}
L → .id, {$,=}
R → to I_7
L → to I_8
* → to I_4
id → to I_5

L: I_2: S → L.=R,$ → to I_6
R → L.,$

R: I_3: S → R.,$

id: I_5: L → id., {$,=}

I_6: S → L=.R,$
R → .L,$
L → .*R,$
L → .id,$
R → to I_9
L → to I_{10}
* → to I_{11}
id → to I_{12}

I_7: L → *R., {$,=}

I_8: R → L., {$,=}

I_9: S → L=R.,$

I_{10}: R → L.,$

I_{11}: L → *.R,$
R → .L,$
L→ .*R,$
L → .id,$
R → to I_{13}
L → to I_{10}
* → to I_{11}
id → to I_{12}

I_{12}: L → id.,$

I_{13}: L → *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'.
   $C \leftarrow \{I_0, \ldots, I_n\}$
2. Create the parsing action table as follows
   - If a is a terminal, $A \rightarrow \alpha . a\beta, b$ in I_i and $goto(I_i, a) = I_j$ then action[i,a] is ***shift j.***
   - If $A \rightarrow \alpha ., a$ is in I_i, then action[i,a] is ***reduce*** $A \rightarrow \alpha$ where $A \neq S'$.
   - If $S' \rightarrow S ., \$$ is in I_i, then 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 $goto(I_i, A) = I_j$ then 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

$\Downarrow$

so, it is a LR(1) grammar