LR Parser Table

Dr. Meera Thapar Khanna

CSE Department

Pandit Deendayal Energy University

Example

Grammar

S -> AA

A -> aA | b

Step 1:

Grammar Augmentation

- S’ -> .S … Rule 0
- S -> .AA … Rule 1
- A -> .aA ….Rule 2
- A -> .b …. Rule 3

Step 2

Closure operation = I0

- S’ -> .S
- S -> .AA
- A -> .aA
- A -> .b

Goto (I0, S) = I1

- S’ -> S. //**

Goto(I0,b) = I4

A -> b. //**

Goto(I0, A) = I2

- S -> A.A
- A -> .aA
- A -> .b

Goto(I2, A) = I5

S -> AA.

Goto (I2, a) = I3

Goto (I2, b) = I4

Goto(I0, a) = I3

- A -> a.A
- A -> .aA
- A -> .b

Goto (I3, A) = I6

A -> aA.

Goto (I3, a) = I3

Goto (I3, b) = I4

Example

Grammar

E -> E + T | T

T -> T * F | F

F -> (E) | id

G:

E -> E + T	(1)
E -> T	(2)
T -> T * F	(3)
T -> F	(4)
F -> (E)	(5)
F -> id	(6)

Step1

Grammar Augumentation G‘:

E‘ -> E	(0)
E -> E + T	(1)
E -> T	(2)
T -> T * F	(3)
T -> F	(4)
F -> (E)	(5)
F -> id	(6)

LR(0) items

Closure of E’ -> .E = i0

i(0): E’-> .E

E -> .E + T

E -> .T

T -> .T * F

T -> .F

F -> .(E)

F -> .id

Step 3

GOTO(i0,E) = i1

GOTO(i0,T) = i2

GOTO(i0, F) =i3

GOTO(i0, () = i4

GOTO(i0, id) = i5

GOTO(i1, +) = i6

GOTO(i2, *) = i7

GOTO(i4, E) = i8

GOTO(i4, T)=i2

GOTO(i4, F)=i3

GOTO(i4, ( )=i4

GOTO(i4,id) =i5

GOTO(i6,T) = i9

GOTO(i6,F)=i3

GOTO(i6,( )=i4

GOTO(i6, id) = i5

GOTO(i7,F)=i10

GOTO(i7,( )=i4

GOTO(i7,id) =i5

GOTO(i8,))=i11

GOTO(i8,+)=i6

GOTO(i9,*)=i7

Rules for construction of parsing table LR(0) from Canonical collections of LR(0) items

Action part: For Terminal Symbols

- If A -> α.aβ is state Ix in Items and goto(Ix,a)=Iy then set action [Ix,a]=Sy (represented as shift to state Iy]
- If A -> α. is in Ix, then set action[Ix, f] to reduce A -> α for all symbols ‘f’.
- If S’ -> S. is in Ix then set action[Ix,$]=accept.

Go To Part: For Non Terminal Symbols

- If goto(Ix, A) = Iy, then goto(Ix,A) in table = Y
- It is numeric value of state Y.
- All other entries are considered

as error.

- Initial state is S’ -> .S

Parsing Table LR(0)

states	id	+	*	(	)	$	E	T	F
	Action Part						GOTO Part		
I0	s5			s4			1	2	3
I1		s6				acc			
I2	r2	r2	s7/r2	r2	r2	r2			
I3	r4	r4	r4	r4	r4	r4			
I4	s5			s4			8	2	3
I5	r6	r6	r6	r6	r6	r6			
I6	s5			s4				9	3
I7	s5			s4					10
I8		s6			s11				
i9	r1	r1	r1/s7	r1	r1	r1			
I10	r3	r3	r3	r3	r3	r3			
I11	r5	r5	r5	r5	r5	r5			

SLR(1)

- SLR(1) Parser is used for accepting the certain grammar which is not accepted by LR(0) parser. SLR(1) Parser stands for Simple LR(1).
- SLR(1) parsers use the same LR(0) Sets and have the Same Table Structure and Parser Operation,
- The difference in SLR(1) Parser with LR(0) Parser comes in Assigning Table Actions.
- SLR(1) Parsers use one token of look-ahead to eliminate the conflicts.
- The Simple Improvement that SLR(1) makes on the basic LR(0) parser is to reduce only if the next input token is a member of the Follow Set of the non-terminal being reduced.

Rules for construction of parsing table SLR(1) from Canonical collections of LR(0) items

Action part: For Terminal Symbols

- If A -> α.aβ is state Ix in Items and goto(Ix,a)=Iy then set action [Ix,a]=Sy (represented as shift to state Iy]
- If A -> α. is in Ix, then set action[Ix, f] to reduce A -> α for all symbols ‘f’ where ‘f’ is in Follow(A)
- If S’ -> S. is in Ix then set action[Ix,$]=accept.

Go To Part: For Non Terminal Symbols

- If goto(Ix, A) = Iy, then goto(Ix,A) in table = Y
- It is numeric value of state Y.
- All other entries are considered

as error.

- Initial state is S’ -> .S

Note: SLR(1) parser avoids the above shift / reduce conflicts by reducing only if the next input token is a member of the Follow Set of the nonterminal being reduced. So the above grammar is not LR(0), but it's SLR(1).

Follow Set

- Follow(E) = {$,), +}
- Follow(T) = {$, +,), *}
- Follow(F) ={*, +,), $}

Parsing Table SLR(1)

states	id	+	*	(	)	$	E	T	F
	Action Part						GOTO Part		
I0	s5			s4			1	2	3
I1		s6				acc			
I2		r2	s7		r2	r2			
I3		r4	r4		r4	r4			
I4	s5			s4			8	2	3
I5		r6	r6		r6	r6			
I6	s5			s4				9	3
I7	s5			s4					10
I8		s6			s11				
i9		r1	s7		r1	r1			
I10		r3	r3		r3	r3			
I11		r5	r5		r5	r5			

Parsing Table: No multiple entries: LR(0) Grammar

Example

Grammar

S -> AA

A -> aA | b

Follow Set

- Follow(S) = $
- Follow(A) = {a,b,$}

Parsing Table SLR(1)

	a	b	$	S	A
	ACTION			GOTO	
I0	S3	S4		1	2
I1			Accept		
I2	S3	S4			5
I3	S3	S4			6
I4	r3	r3	r3		
I5			r1		
I6	r2	r2	r2		

Parsing Table: No multiple entries: LR(0) Grammar