

UNIVERSITY OF ASIA PACIFIC

Department of Computer Science & Engineering

Assignment

Course Code : CSE 429

Course Title : Compiler Design

Submitted by:

Name : Sheikh Nafez Sadnan

Reg. No. : 20101106

Roll No. : 106

Section : B

Submitted To:

Baivab Das

Lecurer

Department of CSE

University of Asia Pacific

LL(1) parser

CFG	First	Follow
$S \rightarrow L = R \mid a b R \mid \epsilon$	$\{*, id, a, \epsilon\}$	$\{\$, \}$
$L \rightarrow * R \mid id$	$\{*, id\}$	$\{\$, =\}$
$R \rightarrow L \mid \epsilon$	$\{*, id, \epsilon\}$	$\{\$, =\}$

Parsing table:

	=	a	b	*	id	\$
S		$S \rightarrow a b R$		$S \rightarrow L = R$ R	$S \rightarrow L = R$	$S \rightarrow \epsilon$
L				$L \rightarrow * R$	$L \rightarrow id$	
R	$R \rightarrow \epsilon$			$R \rightarrow L$	$R \rightarrow L$	$R \rightarrow \epsilon$

There is no multiple entry in a single cell so, the CFG is LL(1).

Given input strings $*id = id \$$

Stack	Input	Action
$\$ S$	$* id = id \$$	$S \rightarrow L = R$
$\$ R = L$	$* id = id \$$	$L \rightarrow * R$
$\$ R = R *$	$* id = id \$$	Match & pop
$\$ R = R$	$id = id \$$	$R \rightarrow L$
$\$ R = L$	$id = id \$$	$L \rightarrow id$
$\$ R = id$	$id = id \$$	Match & pop
$\$ R =$	$= id \$$	Match & pop
$\$ R$	$id \$$	$R \rightarrow L$
$\$ L$	$id \$$	$L \rightarrow id$
$\$ id$	$id \$$	Match & pop
$\$$	$\$$	Accept

so the input string is parsed.

SLR(1) parser

CFG	First	Follow
$S \rightarrow L = R \mid a b R \mid \epsilon$	$\{*, id, a, \epsilon\}$	$\{ \}$
$L \rightarrow * R \mid id$	$\{*, id\}$	$\{ \}, \{ \}$
$R \rightarrow L \mid \epsilon$	$\{*, id, \epsilon\}$	$\{ \}, \{ \}$

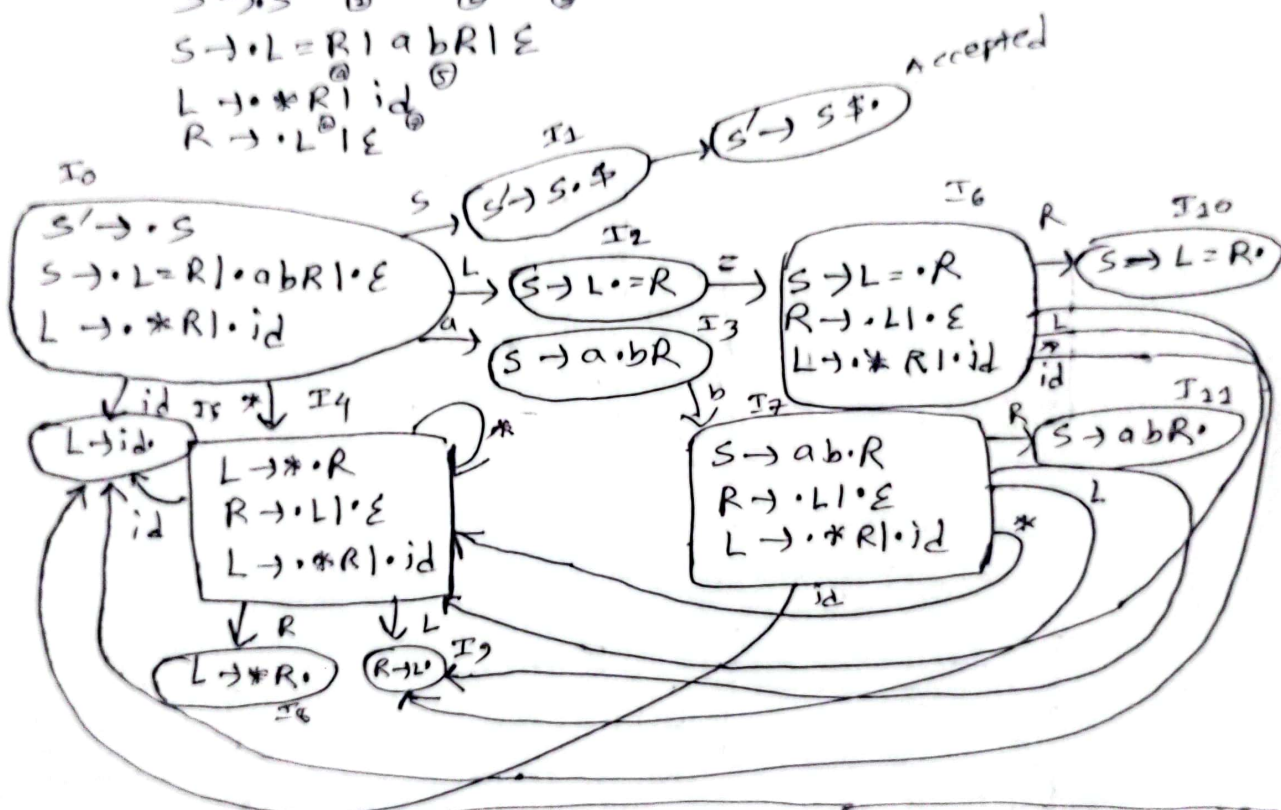
Here,

$S' \rightarrow S$ ① ② ③

$S \rightarrow \cdot L = R \mid a b R \mid \epsilon$

$L \rightarrow \cdot * R \mid id$ ④ ⑤

$R \rightarrow \cdot L \mid \epsilon$ ⑥



Parsing table		Action						Goto		
State		ϵ	a	b	*	id	$\$$	S	L	R
0			S3		S4	S5	r3	1	2	
1							Accepted			
2		S6								
3				S7						
4		r7			S4	S5	r7		9	8
5		r5					r5			
6		r7			S4	S5	r7		9	
7		r7			S4	S5	r7		9	10
8		r4					r4			11
9		r6					r6			
10							r2			
11							r2			

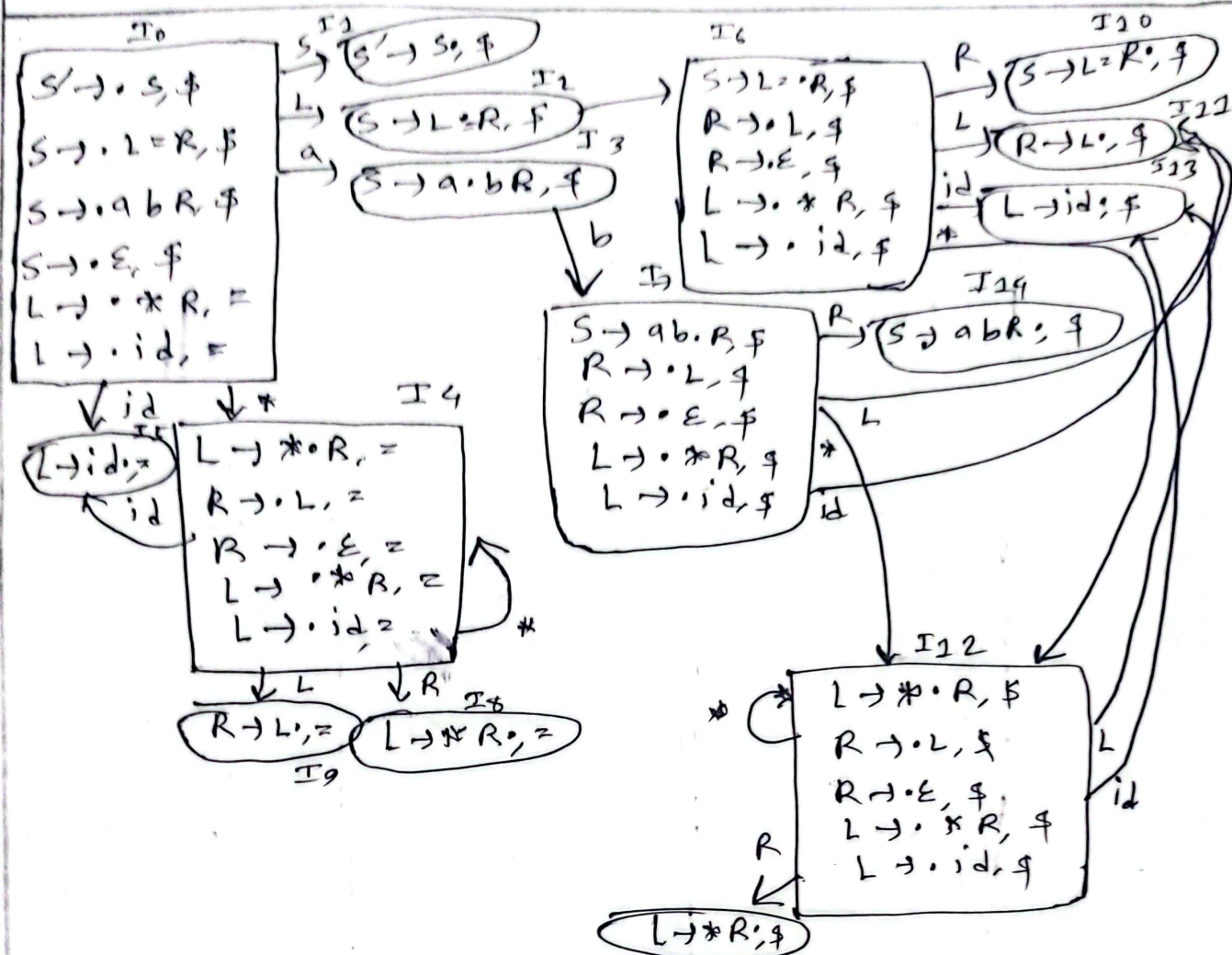
There is no conflict because there is no multiple entries in single cell. So it is SLR(1).
 Input string $\rightarrow *id=id$

Stack	Input	Action
0	*id=id\$	Shift 4
0*4	id=id\$	Shift 5
0*4id5	=id\$	Reduce $L \rightarrow id$
0*4L2	=id\$	Reduce $R \rightarrow L$
0*4R6	=id\$	Reduce $L \rightarrow *R$
0L2	=id\$	Shift 6
0L2=6	id\$	Shift 5
0L2=6id5	\$	Reduce $L \rightarrow id$
0L2=6L2	\$	Reduce $R \rightarrow L$
0L2=6R10	\$	Reduce $S \rightarrow L=R$
0S1	\$	Accept

So, the input string is parsed
 :: :: CLR(1) Parser

CFG	First	Follow
$S \rightarrow L=R \mid a b R \mid \epsilon$	$\{*, id, a, \epsilon\}$	$\{ \$ \}$
$L \rightarrow *R \mid id$	$\{*, id\}$	$\{ \$, = \}$
$R \rightarrow L \mid \epsilon$	$\{*, id, \epsilon\}$	$\{ \$, = \}$

$S' \rightarrow S$ ① ②
 $S \rightarrow L=R \mid a b R \mid \epsilon$ ③
 $L \rightarrow *R \mid id$ ④ ⑤
 $R \rightarrow L \mid \epsilon$ ⑥ ⑦



Parsing table of CLR(0) parser: I25

State	Action						Goto		
	=	a	b	*	id	\$	S	L	R
0		S3		S4	S5	r3	1	2	
1						Accept			
2	S6								
3			S7						
4	r7			S4	S5			9	8
5	r5								
6				S12	S13	r7		11	10
7				S12	S13			11	14
8	r4								
9	r6								
10						r1			
11						r6			
12				S12	S13	r7		11	15
13						r5			
14						r2			
15						r4			

There is no conflict in the parsing table because there is no multiple entries in a single cell of the table. So it is CLR(1),
input string $\rightarrow *id=id$,

Stack	Input	Action
0	* id = id \$	shift 4
0*4	id = id \$	shift 5
0*4id5	= id \$	reduce $L \rightarrow id$
0*4L9	= id \$	reduce $R \rightarrow L$
0*4R8	= id \$	reduce $L \rightarrow *R$
0L2	= id \$	shift 6
0L2=6	id \$	shift 13
0L2=6id13	\$	reduce $L \rightarrow id$
0L2=6L12	\$	reduce $R \rightarrow L$
0L2=6R20	\$	reduce $S \rightarrow L=R$
0S1	\$	Accept

The input string is parsed with CLR(1) parser.
LALR(1) parser

$S \rightarrow \cdot L = R \mid a b R \mid \epsilon$

$L \rightarrow \cdot * R \mid id$

$R \rightarrow \cdot L \mid \epsilon$

Here, we can observe (I_4, I_{12}) , (I_5, I_{13}) ,
 (I_7, I_{11}) , (I_6, I_{15}) has same production
So, we merge those states.

Parse table of LALR(1):

State	Action						Goto		
	=	a	b	*	id	\$	S	L	R
0		S3		S412	S523	P3	1	2	
1						Accept			
2	S6								
3			S7						
412	P7			S412	S523	P7		211	815
513	P5					P5			
6				S412	S523	P7		911	10
7				S412	S523			911	14
815	P4					P4			
911	P6					P6			
10						P1			
14						P2			

No multiple entries, so the CFG is LALR(1)

Input string: *id=id

Stack	Input	Action
0	*id=id	Shift 412
0*412	id=id	Shift 523
0*412id523	=id	Reduce L → id
0*412L911	=id	Reduce R → L
0*412R815	=id	Reduce L → *R
DL2	=id	Shift 6
DL2=6	id	Shift 523
DL2=6id523		Reduce L → id
DL2=6L911		Reduce R → L
DL2=6R10		Reduce S → L = R
OS1		Accept

So, it can be said that input string *id=id is parsed with LALR(1) parser.