دانشگاه صنعتی اصفهان

دانشکده مهندسی برق و کامپیوتر

درس کامپایلر

پاسخ تکلیف تئوری سوم

Q1)

A) $LR(0) \subset LALR(1) \subset LR(2) \subset LR(1)$

wrong, LR(2) is more powerful than LR(1)

for example, the grammar below is not LR(1) but it is LR(2)

$$S \rightarrow A1\ B\ x \mid A2\ B\ y$$
$$A1 \rightarrow a$$
$$A2 \rightarrow a$$
$$B \rightarrow b$$

B) any CFG can be parsed in $O(n^3)$

correct, all CFGs can be parsed using CYK in O(n^3)

C) all unambiguous grammars can be parsed using $LR(1)$

wrong, the grammar $A \rightarrow aAa \mid \varepsilon$ is not ambiguous but also can't be parsed using LR(1) because of conflict

Q2)

Power:

$$SLR(1) \subset LALR(1) \subset LR(1)$$

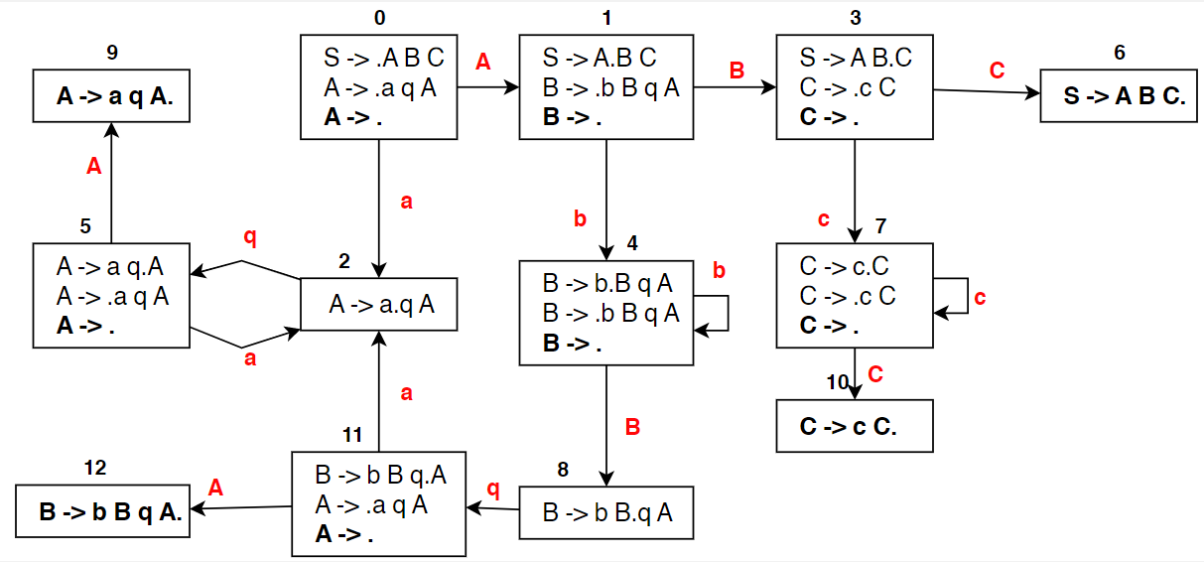
Number of states:

$$LR(1) > LALR(1) = SLR(1)$$

Q3) consider the following grammar

$$\mathbf{1}: S \rightarrow A\ B\ C$$
$$\mathbf{2-3}: A \rightarrow a\ q\ A \mid \epsilon$$
$$\mathbf{4-5}: B \rightarrow b\ B\ q\ A \mid \epsilon$$
$$\mathbf{6-7}: C \rightarrow c\ C \mid \epsilon$$

A) draw the $\boldsymbol{SLR(1)}$ **state automata** and build the **parsing table** of this grammar



SLR table									
State	ACTION					GOTO			
	a	q	b	c	$	S	A	B	C
0	s2	r_2	r_2	r_2	r_2		1		
1		r_4	s4	r_4	r_4			3	
2		s5							
3				s7	r_6				6
4		r_4	s4	r_4	r_4			8	
5	s2	r_2	r_2	r_2	r_2		9		
6					acc				
7				s7	r_6				10
8		s11							
9		r_1	r_1	r_1	r_1				
10					r_5				
11	s2	r_2	r_2	r_2	r_2		12		
12		r_3		r_3	r_3				

B) **parse** the following input and show the step-by-step status of **stack**, **input line** and **actions taken**

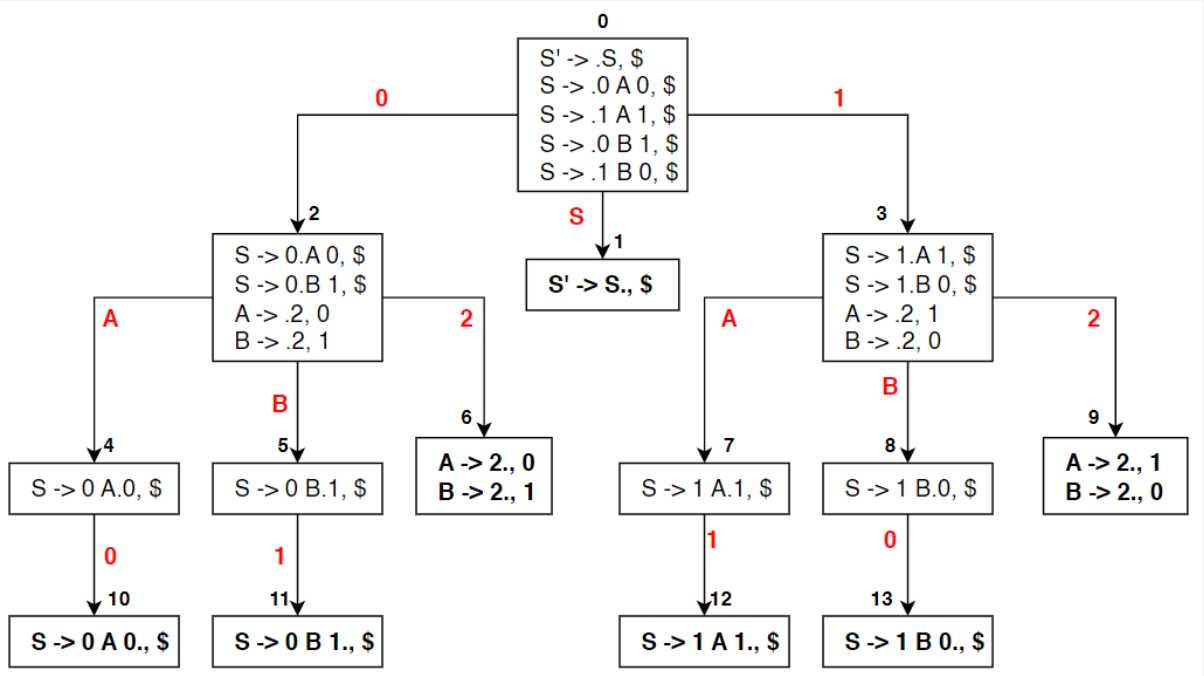
$$a\ q\ b\ q\ c\ c\ \$$$

Trace			
Step	**Stack**	**Input**	**Action**
1	0	a q b q c c $	s2
2	0 2	q b q c c $	s5
3	0 2 5	b q c c $	r_2
4	0 2 5	b q c c $	9
5	0 2 5 9	b q c c $	r_1
6	0	b q c c $	1
7	0 1	b q c c $	s4
8	0 1 4	q c c $	r_4
9	0 1 4	q c c $	8
10	0 1 4 8	q c c $	s11
11	0 1 4 8 11	c c $	r_2
12	0 1 4 8 11	c c $	12
13	0 1 4 8 11 12	c c $	r_3
14	0 1	c c $	3
15	0 1 3	c c $	s7
16	0 1 3 7	c $	s7
17	0 1 3 7 7	$	r_6
18	0 1 3 7 7	$	10
19	0 1 3 7 7 10	$	r_5
20	0 1 3 7	$	10
21	0 1 3 7 10	$	r_5
22	0 1 3	$	6
23	0 1 3 6	$	acc

Q4) consider the following grammar

$$S' \rightarrow S$$
$$S \rightarrow 0A0 \mid 1A1 \mid 0B1 \mid 1B0$$
$$A \rightarrow 2$$
$$B \rightarrow 2$$

A) draw the $\boldsymbol{LR(1)}$ **state automata**



B) build the $\boldsymbol{LR(1)}$ **parsing table** of this grammar

LR table								
State	ACTION				GOTO			
	0	1	2	$	S'	S	A	B
0	s2	s3				1		
1				acc				
2			s6				4	5
3			s9				7	8
4	s10							
5		s11						
6	r_5	r_6						
7		s12						
8	s13							
9	r_6	r_5						
10				r_1				
11				r_3				
12				r_2				
13				r_4				

C) is this grammar $\boldsymbol{LALR(1)}$?

If not, briefly describe **why**, otherwise combine states from part A to build the $LALR(1)$ **automata**.

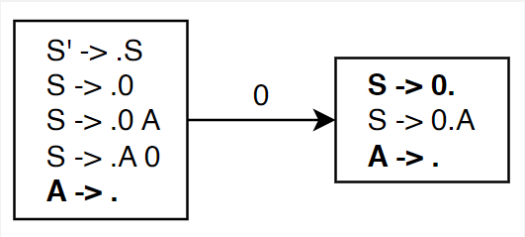
In order to construct the $LALR(1)$ automata of this grammar, we need to combine states 6 and 9, which results in a reduce/reduce conflict of r5 and r6

Q5) Over the alphabet {0}, provide a grammar that…

A)

has **exactly one** shift/reduce conflict and **exactly one** reduce/reduce conflict in $\boldsymbol{SLR(1)}$. Also, provide the sets and GOTO transitions of **state automata (starting from set 0)** that reaches those conflicts (you don't need to draw the whole automata)

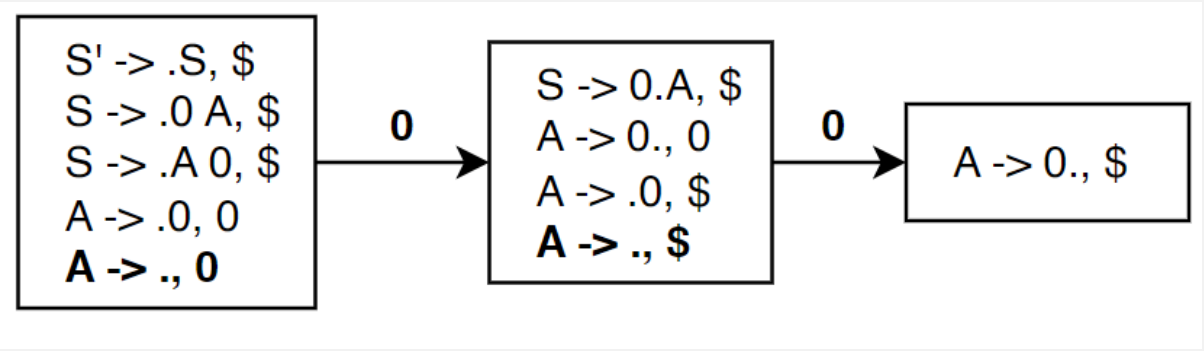
$$S \rightarrow 0 \mid 0A \mid A0$$
$$A \rightarrow \epsilon$$



B)

has **exactly 2** shift/reduce conflicts and **no** reduce/reduce conflict in $\boldsymbol{LR(1)}$. Also, provide the sets and GOTO transitions of **state automata (starting from set 0)** that reaches those conflicts (you don't need to draw the whole automata)

$$S \rightarrow 0A|A0$$
$$A \rightarrow 0 \mid \epsilon$$



Q6) Below is the $SLR(1)$ parsing table of a grammar, along with the size of each production. find the whole grammar, explain with detail.

SLR table											
State	ACTION						GOTO				
	0	1	a	b	c	$	S'	S	A	B	C
0			s4		s3			1	2		
1						acc					
2	s5										
3						r_2					
4	r_4		s4						6		
5				s8						7	
6	r_3										
7		s9		s10							
8		r_6		r_6							
9					s12						11
10		r_5		r_5							
11						r_1					
12						r_7					

#	LHS	Size of RHS
0	S’	1
1	S	5
2	S	1
3	A	2
4	A	1
5	B	2
6	B	1
7	C	1

Terminals: $0 \quad 1 \quad a \quad b \quad c \quad \$$

Non terminals: S A B C

start symbol: S’

based on GOTO(0,S) = 1 and acc in state 1, we can fill #0 rule:

$$S' \rightarrow S$$

With that rule found, if we try to compute the closure in state 0, we get S rules

$$S' \rightarrow .S$$
$$S \rightarrow .X$$
$$S \rightarrow .XXXXX$$

The GOTO functions for these rules are as follow:

GOTO(0,A) = 2 GOTO(0,c) = 3 GOTO(0,a) = 4

if we look at state 3, we can see reduction with rule #2, combined with the GOTO(0,c) = 3, we can fill rule #2:

$$S \to c$$

Based on the fact that we have two GOTOs remaining, but one rule, we can come to conclusion that the first symbol of the remaining rule is A, so we get another closure and hence, we can find the other GOTO function:

$$S' \to .S$$
$$S \to .c$$
$$S \to .AXXXX$$
$$A \to .XX$$
$$A \to .X$$

Based on the only remining GOTO(0,a) = 4, we can fill the blanks (also, state 4 has reduction with rule 4, that's also another clue!):

$$S' \to .S$$
$$S \to .c$$
$$S \to .AXXXX$$
$$A \to .aX$$
$$A \to .a$$

Let's go to state 4, we have shift 4 (recursion) and also GOTO(0,A) = 6, we also have reduction with rule 3 in state 6, so that fills rule #3

$$A \to aA$$

let's go to state 2, in which we have the following kernel:

$$S \to A.XXXX$$

The only action in this state is shift to state 5 with symbol 0, so that's the next symbol.

Let's go to state 5, we get the following kernel:

$$S \to A0.XXX$$

In this state, we have GOTO(5,B) = 7, that gives us the next symbol. Now we compute the closure for state 5:

$$S \to A0.BXX$$
$$B \to .XX$$
$$B \to .X$$

We have GOTO(5,B) = 7, which gives us the next symbol for rule #5, and GOTO(5,b) = 8, and also have reduction with rule #6 in state 8, now we can fill rule #6:

$$B \to b$$

Let's go to state 7, in which we have:

$$S \to A0B.XX$$
$$B \to B.X$$

In this state, we have GOTO(7,1) = 9 and GOTO(7,b) = 10, and we only have reduction with rule #5 in state 10, that gives us:

$$B \to Bb$$

Let's visit state 9:

$$S \to A0B1.X$$

Finally! We found non-terminal C in GOTO(9,C) = 11, that gives us this closure:

$$S \to A0B1.C$$
$$C \to .X$$

We have GOTO(9,c) = 12 and reduction with rule #7 in state 12, that's the final missing piece

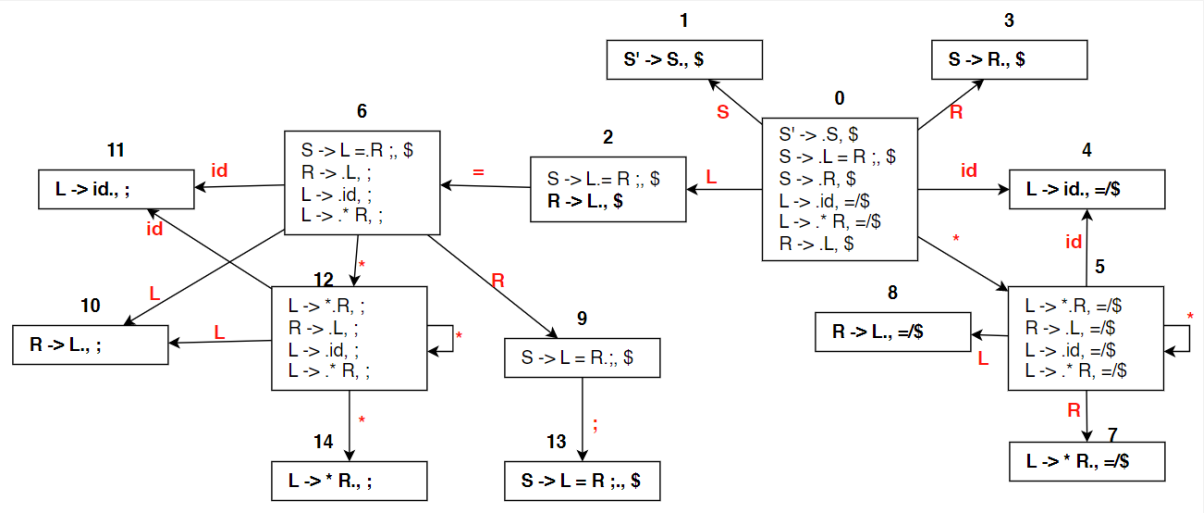
The whole answer put together is:

$$S' \to S$$
$$S \to A0B1C \mid c$$
$$A \to aA \mid a$$
$$B \to Bb \mid b$$
$$C \to c$$

Q7) Consider the following grammar

$$S' \rightarrow S$$
$$S \rightarrow L = R\ ; \mid R$$
$$L \rightarrow id \mid * R$$
$$R \rightarrow L$$

A) draw its $LR(1)$ **state automata**



B) using the answer for part A, tell which states should be **merged** together to get to $\boldsymbol{LALR(1)}$

States 8 and 10, States 7 and 14, States 5 and 12, States 4 and 11

C) construct $LALR(1)$ **parsing table**

LR table									
State	ACTION					GOTO			
	=	;	id	*	$	S'	S	L	R
0			s4	s5			1	2	3
1					acc				
2	s6				r_5				
3					r_2				
4	r_3	r_3			r_3				
5			s4	s5				8	7
6			s4	s5				8	9
7	r_4	r_4			r_4				
8	r_5	r_5			r_5				
9		s10							
10					r_1				

D) **parse** the following input and show the step-by-step status of **stack**, **input line** and **actions taken**

$$id = id\ ;$$

Trace			
Step	**Stack**	**Input**	**Action**
1	0	id = id ; $	s4
2	0 4	= id ; $	r_3
3	0	= id ; $	2
4	0 2	= id ; $	s6
5	0 2 6	id ; $	s4
6	0 2 6 4	; $	r_3
7	0 2 6	; $	8
8	0 2 6 8	; $	r_5
9	0 2 6	; $	9
10	0 2 6 9	; $	s10
11	0 2 6 9 10	$	r_1
12	0	$	1
13	0 1	$	acc

E) show that this grammar is **not $SLR(1)$,** you don't need to draw the whole state diagram, start from state 0 and reach a conflict.