German University in Cairo
Department of Computer Science
Assoc. Prof. Haythem O. Ismail

CSEN 1003 Compiler, Spring Term 2020
Practice Assignment 8

Exercise 8-1

SLR and LALR Parsing

Show that the following grammar is LALR but not SLR.

$$\begin{array}{lcl} S & \rightarrow & A\texttt{a} \mid \texttt{b}A\texttt{c} \mid \texttt{dc} \mid \texttt{bda} \\ A & \rightarrow & \texttt{d} \end{array}$$

Solution:

We first construct the LR(0) Automaton to construct the SLR parsing table.
Augmented grammar:

$$\begin{array}{lcl} S' & \rightarrow & S \\ S & \rightarrow & A\texttt{a} \mid \texttt{b}A\texttt{c} \mid \texttt{dc} \mid \texttt{bda} \\ A & \rightarrow & \texttt{d} \end{array}$$

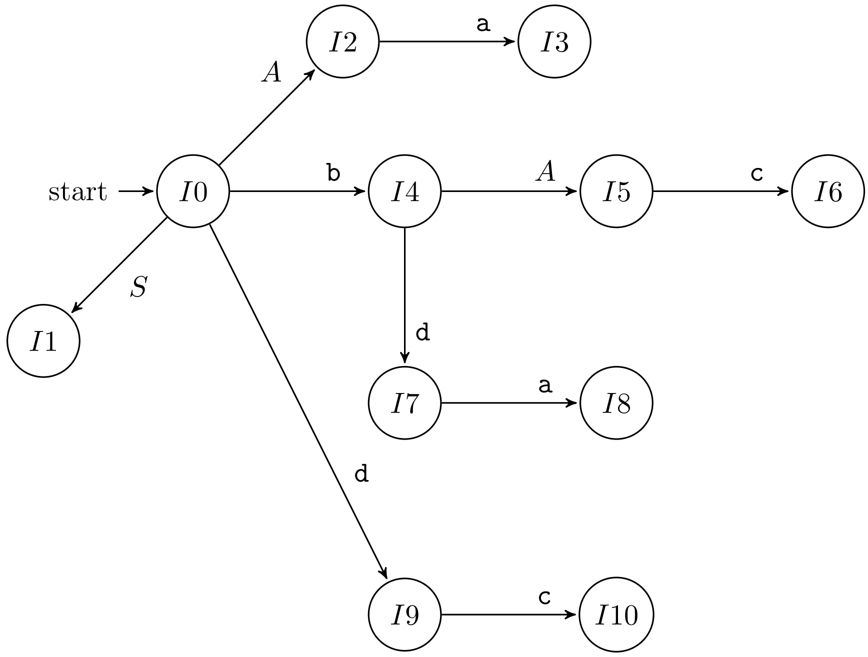
Rule numbering:

1. $S \rightarrow A\texttt{a}$
2. $S \rightarrow \texttt{b}A\texttt{c}$
3. $S \rightarrow \texttt{dc}$
4. $S \rightarrow \texttt{bda}$
5. $A \rightarrow \texttt{d}$

LR(0) Item sets:

I_0: $S' \rightarrow \cdot S$ $S \rightarrow \cdot A\texttt{a}$ $S \rightarrow \cdot\texttt{b}A\texttt{c}$ $S \rightarrow \cdot\texttt{dc}$ $S \rightarrow \cdot\texttt{bda}$ $A \rightarrow \cdot\texttt{d}$	I_1: $S' \rightarrow S\cdot$	I_2: $S \rightarrow A \cdot \texttt{a}$
I_3: $S \rightarrow A\texttt{a}\cdot$	I_4: $S \rightarrow \texttt{b} \cdot A\texttt{c}$ $S \rightarrow \texttt{b} \cdot \texttt{da}$ $A \rightarrow \cdot\texttt{d}$	I_5: $S \rightarrow \texttt{b}A \cdot \texttt{c}$ I_6: $S \rightarrow \texttt{b}A\texttt{c}\cdot$
I_7: $S \rightarrow \texttt{bd} \cdot \texttt{a}$ $A \rightarrow \texttt{d}\cdot$	I_8: $S \rightarrow \texttt{bda}\cdot$	I_9: $S \rightarrow \texttt{d} \cdot \texttt{c}$ $A \rightarrow \texttt{d}\cdot$ I_{10}: $S \rightarrow \texttt{dc}\cdot$

LR(0) Automaton:



SLR Parsing Table:

State	Action					GOTO	
	a	b	c	d	$	S	A
0		s4	s9			1	2
1					accept		
2	s3						
3					r1		
4				s7			5
5			s6				
6					r2		
7	s8,r5		r5				
8					r4		
9	r5		s10,r5				
10					r3		

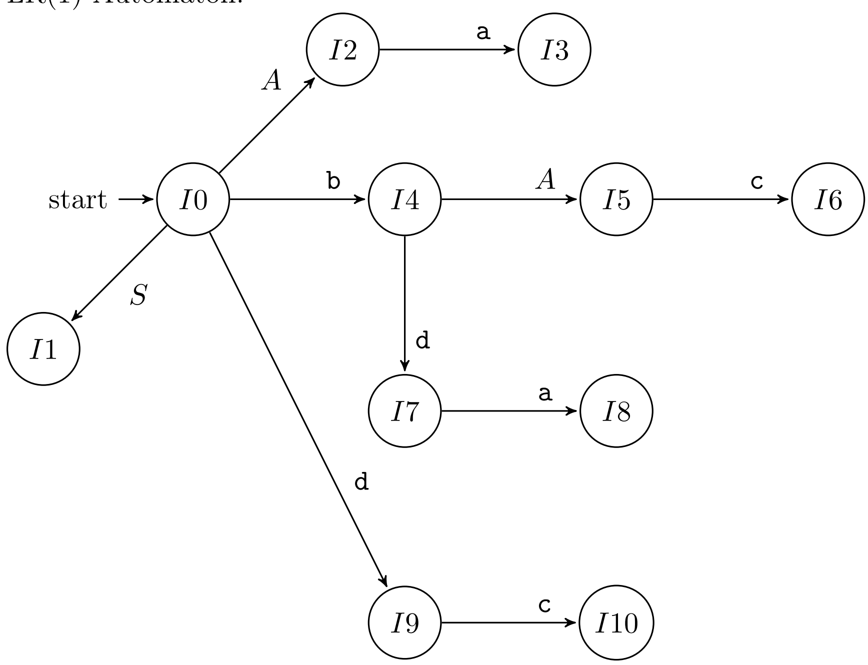
The table above shows that the grammar is *not* SLR since there is a shift/reduce conflict in states I_7 and I_9.

To construct the LALR automaton, we first construct the LR(1) automaton and combine the core equivalent states.

LR(1) Item sets:

I_0: $S' \to \cdot S, \$$ $S \to \cdot A\texttt{a}, \$$ $S \to \cdot \texttt{b}A\texttt{c}, \$$ $S \to \cdot \texttt{dc}, \$$ $S \to \cdot \texttt{bda}, \$$ $A \to \cdot \texttt{d}, a$	I_1: $S' \to S\cdot, \$$	I_2: $S \to A \cdot \texttt{a}, \$$
I_3: $S \to A\texttt{a}\cdot, \$$	I_4: $S \to \texttt{b} \cdot A\texttt{c}, \$$ $S \to \texttt{b} \cdot \texttt{da}, \$$ $A \to \cdot \texttt{d}, c$	I_5: $S \to \texttt{b}A \cdot \texttt{c}, \$$ I_6: $S \to \texttt{b}A\texttt{c}\cdot, \$$
I_7: $S \to \texttt{bd} \cdot \texttt{a}, \$$ $A \to \texttt{d}\cdot, c$	I_8: $S \to \texttt{bda}\cdot, \$$	I_9: $S \to \texttt{d} \cdot \texttt{c}, \$$ $A \to \texttt{d}\cdot, a$ I_{10}: $S \to \texttt{dc}\cdot, \$$

LR(1) Automaton:



Since the LR(1) automaton has no core-equivalent states, then the LALR automaton is the same as the LR(1) automaton.

LR(1)/LALR Parsing Table:

State	Action					GOTO	
	a	b	c	d	$	S	A
0		s4	s9			1	2
1					accept		
2	s3						
3					r1		
4				s7			5
5			s6				
6					r2		
7	s8		r5				
8					r4		
9	r5		s10				
10					r3		

The grammar is LR(1)/LALR since the above table has no conflicts.

Exercise 8-2

LALR and LR(1) Parsing

Show that the following grammar is LR(1) but not LALR.

$$\begin{array}{lcl} S & \rightarrow & A\texttt{a} \mid \texttt{b}A\texttt{c} \mid B\texttt{c} \mid \texttt{b}B\texttt{a} \\ A & \rightarrow & \texttt{d} \\ B & \rightarrow & \texttt{d} \end{array}$$

Solution:

Augmented grammar:

$$\begin{array}{lcl} S' & \rightarrow & S \\ S & \rightarrow & A\texttt{a} \mid \texttt{b}A\texttt{c} \mid B\texttt{c} \mid \texttt{b}B\texttt{a} \\ A & \rightarrow & \texttt{d} \\ B & \rightarrow & \texttt{d} \end{array}$$

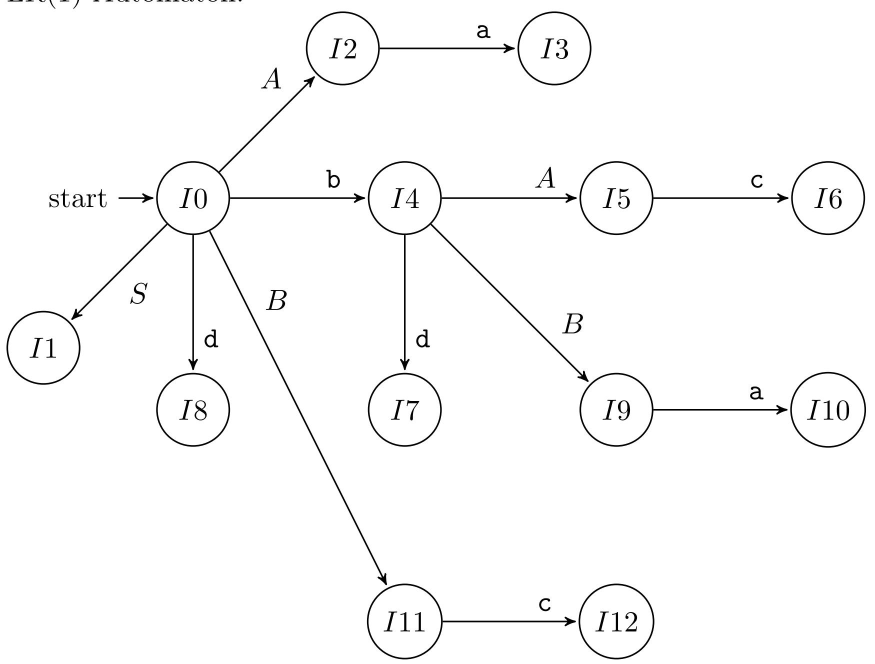
Rule numbering:

1. $S \rightarrow A\texttt{a}$
2. $S \rightarrow \texttt{b}A\texttt{c}$
3. $S \rightarrow B\texttt{c}$
4. $S \rightarrow \texttt{b}B\texttt{a}$
5. $A \rightarrow \texttt{d}$
6. $B \rightarrow \texttt{d}$

LR(1) Item sets:

I_0: $S' \rightarrow \cdot S, \$$ $S \rightarrow \cdot A\texttt{a}, \$$ $S \rightarrow \cdot \texttt{b}A\texttt{c}, \$$ $S \rightarrow \cdot B\texttt{c}, \$$ $S \rightarrow \cdot \texttt{b}B\texttt{a}, \$$ $A \rightarrow \cdot \texttt{d}, a$ $B \rightarrow \cdot \texttt{d}, c$	I_1: $S' \rightarrow S\cdot, \$$	I_2: $S \rightarrow A \cdot \texttt{a}, \$$ I_3: $S \rightarrow A\texttt{a}\cdot, \$$
I_4: $S \rightarrow \texttt{b} \cdot A\texttt{c}, \$$ $S \rightarrow \texttt{b} \cdot B\texttt{a}, \$$ $A \rightarrow \cdot \texttt{d}, c$ $B \rightarrow \cdot \texttt{d}, a$	I_5: $S \rightarrow \texttt{b}A \cdot \texttt{c}, \$$ I_6: $S \rightarrow \texttt{b}A\texttt{c}\cdot, \$$	I_7: $A \rightarrow \texttt{d}\cdot, c$ $B \rightarrow \texttt{d}\cdot, a$
I_8: $A \rightarrow \texttt{d}\cdot, a$ $B \rightarrow \texttt{d}\cdot, c$	I_9: $S \rightarrow \texttt{b}B \cdot \texttt{a}, \$$ I_{10}: $S \rightarrow \texttt{b}B\texttt{a}\cdot, \$$	I_{11}: $S \rightarrow B \cdot \texttt{c}, \$$ I_{12}: $S \rightarrow B\texttt{c}\cdot, \$$

LR(1) Automaton:



LR(1) Parsing Table:

State	Action					GOTO		
	`a`	`b`	`c`	`d`	$	S	A	B
0		s4		s8		1	2	11
1					accept			
2	s3							
3					r1			
4				s7			5	9
5			s6					
6					r2			
7	r6		r5					
8	r5		r6					
9	s10							
10					r4			
11			s12					
12					r3			

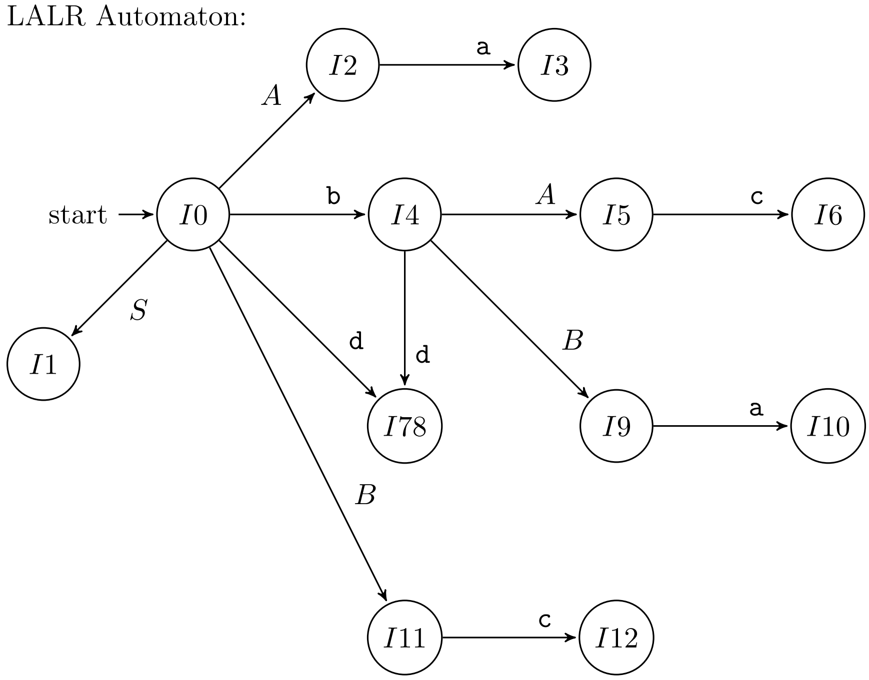
Since the above table has no conflicts, the grammar is LR(1).

To construct the LALR automaton we combine the core equivalent states I_7 and I_8 in the LR(1) automaton.

LALR Item sets:

I_0: $S' \to \cdot S, \$$ $S \to \cdot A\texttt{a}, \$$ $S \to \cdot \texttt{b}A\texttt{c}, \$$ $S \to \cdot B\texttt{c}, \$$ $S \to \cdot \texttt{b}B\texttt{a}, \$$ $A \to \cdot \texttt{d}, a$ $B \to \cdot \texttt{d}, c$	I_1: $S' \to S\cdot, \$$	I_2: $S \to A \cdot \texttt{a}, \$$ I_3: $S \to A\texttt{a}\cdot, \$$
I_4: $S \to \texttt{b} \cdot A\texttt{c}, \$$ $S \to \texttt{b} \cdot B\texttt{a}, \$$ $A \to \cdot \texttt{d}, c$ $B \to \cdot \texttt{d}, a$	I_5: $S \to \texttt{b}A \cdot \texttt{c}, \$$ I_6: $S \to \texttt{b}A\texttt{c}\cdot, \$$	I_{78}: $A \to \texttt{d}\cdot, a/c$ $B \to \texttt{d}\cdot, a/c$
I_9: $S \to \texttt{b}B \cdot \texttt{a}, \$$	I_{10}: $S \to \texttt{b}B\texttt{a}\cdot, \$$	I_{11}: $S \to B \cdot \texttt{c}, \$$ I_{12}: $S \to B\texttt{c}\cdot, \$$

LALR Automaton:



LALR Parsing Table:

State	Action					GOTO		
	a	b	c	d	\$	S	A	B
0		s4		s78		1	2	11
1					accept			
2	s3							
3					r1			
4				s78			5	9
5			s6					
6					r2			
78	r5,r6		r5,r6					
9	s10							
10					r4			
11			s12					
12					r3			

The above grammar is *not* LALR as the parsing table contains reduce/reduce conflicts in state I_{78}.

Exercise 8-3

Canonical LR(1) Parsing

Consider the following grammar:

$$\begin{array}{lcl} S & \rightarrow & X\texttt{a} \\ X & \rightarrow & \texttt{a} \mid \texttt{a}X\texttt{b} \end{array}$$

a) Compute the canonical LR(1) item sets and construct the DFA of the augmented grammar.

Solution:

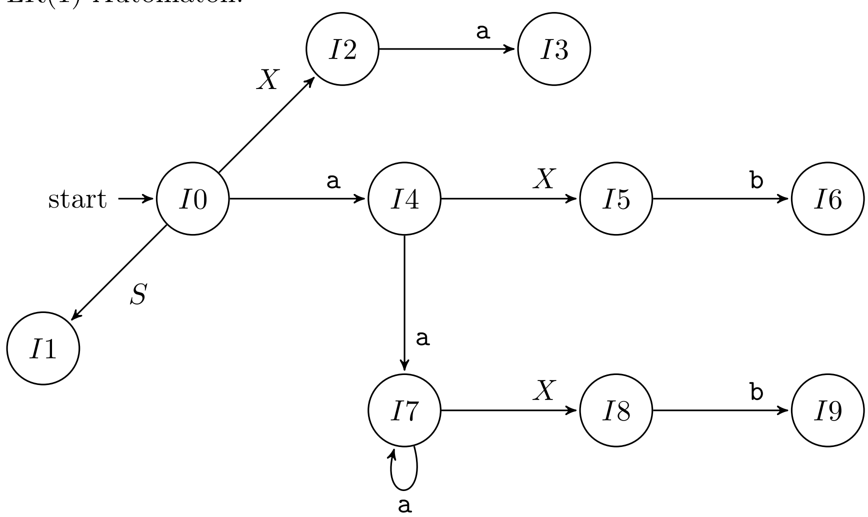
Augmented grammar:

$$\begin{array}{lcl} S' & \rightarrow & S \\ S & \rightarrow & X\texttt{a} \\ X & \rightarrow & \texttt{a} \mid \texttt{a}X\texttt{b} \end{array}$$

LR(1) Item sets:

I_0: $S' \rightarrow \cdot S, \$$ $S \rightarrow \cdot X\texttt{a}, \$$ $X \rightarrow \cdot\texttt{a}, \texttt{a}$ $X \rightarrow \cdot\texttt{a}X\texttt{b}, \texttt{a}$	I_1: $S' \rightarrow S\cdot, \$$	I_2: $S \rightarrow X \cdot \texttt{a}, \$$
I_3: $S \rightarrow X\texttt{a}\cdot, \$$	I_4: $X \rightarrow \texttt{a}\cdot, \texttt{a}$ $X \rightarrow \texttt{a} \cdot X\texttt{b}, \texttt{a}$ $X \rightarrow \cdot\texttt{a}, \texttt{b}$ $X \rightarrow \cdot\texttt{a}X\texttt{b}, \texttt{b}$	I_5: $X \rightarrow \texttt{a}X \cdot \texttt{b}, \texttt{a}$
I_6: $X \rightarrow \texttt{a}X\texttt{b}\cdot, \texttt{a}$	I_7: $X \rightarrow \texttt{a}\cdot, \texttt{b}$ $X \rightarrow \texttt{a} \cdot X\texttt{b}, \texttt{b}$ $X \rightarrow \cdot\texttt{a}X\texttt{b}, \texttt{b}$ $X \rightarrow \cdot\texttt{a}, \texttt{b}$	I_8: $X \rightarrow \texttt{a}X \cdot \texttt{b}, \texttt{b}$ I_9: $X \rightarrow \texttt{a}X\texttt{b}\cdot, \texttt{b}$

LR(1) Automaton:



b) Construct the canonical LR(1) parsing table.

Solution:

Rule numbering:

1. $S \rightarrow X\texttt{a}$
2. $X \rightarrow \texttt{a}$
3. $X \rightarrow \texttt{a}X\texttt{b}$

Parsing table:

State	Action			GOTO	
	`a`	`b`	\$	S	X
0	s4			1	2
1			accept		
2	s3				
3			r1		
4	s7,r2				5
5		s6			
6	r3				
7	s7	r2			8
8		s9			
9		r3			

Note: The table above shows that the grammar is *not* LR(1) since there is a shift/reduce conflict in state I_4.

c) Use the parsing table to simulate canonical LR(1) parsing on the string: `aaba`

Solution:

Stack	Input	Action
0	`aaba`\$	shift
0 4	`aba`\$	shift
0 4 7	`ba`\$	reduce $X \to \mathtt{a}$
0 4 5	`ba`\$	shift
0 4 5 6	`a`\$	reduce $X \to \mathtt{a}X\mathtt{b}$
0 2	`a`\$	shift
0 2 3	\$	reduce $S \to X\mathtt{a}$
0 1	\$	accept