

Computer Science Language Processors

Rules

- The duration of the test is **90 minutes**
- Questions will not be answered during the test
- One cannot re-enter the classroom after leaving it
- The answers must be written using a pen (not a pencil)

1.- Given the grammar:

$$\begin{aligned} S &::= aSD \mid b \\ D &::= Dc \mid e \end{aligned}$$

A) Construct the parsing table for the SLR(1) parser and represent the DFA that is obtained using this methodology.

Augmented Grammar:

- (0) $S' \rightarrow S$
- (1) $S \rightarrow aSD$
- (2) $S \rightarrow b$
- (3) $D \rightarrow Dc$
- (4) $D \rightarrow e$

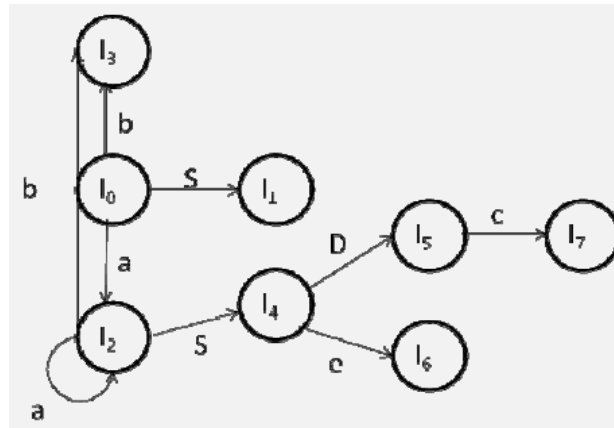
LR(0) canonical set

$I_0 = \{ [S' \rightarrow \cdot S], [S \rightarrow \cdot aSD], [S \rightarrow \cdot b] \}$
 $I_1 = \text{goto}(I_0, S) = \{ [S' \rightarrow S \cdot] \}$
 $I_2 = \text{goto}(I_0, a) = \{ [S \rightarrow a \cdot SD], [S \rightarrow \cdot aSD], [S \rightarrow \cdot b] \}$
 $I_3 = \text{goto}(I_0, b) = \{ [S \rightarrow b \cdot] \}$
 $I_4 = \text{goto}(I_2, S) = \{ [S \rightarrow aS \cdot D], [D \rightarrow \cdot Dc], [D \rightarrow \cdot e] \}$

$\text{goto}(I_2, a) = I_2$
 $\text{goto}(I_2, b) = I_3$

$I_5 = \text{goto}(I_4, D) = \{ [S \rightarrow aSD \cdot], [D \rightarrow D \cdot c] \}$
 $I_6 = \text{goto}(I_4, e) = \{ [D \rightarrow e \cdot] \}$
 $I_7 = \text{goto}(I_5, c) = \{ [D \rightarrow Dc \cdot] \}$

We can represent the DFA as follows:



Finally, to construct the SLR(1) parsing table we need to know:

FOLLOW(S) = {e, \$}
 FOLLOW(D) = {e, c, \$}

STATES	ACTIONS					GOTO	
	a	b	c	e	\$	S	D
0	S2	S3				1	
1					ACCEPT		
2	S2	S3				4	
3				R2	R2		
4				S6			5
5			S7	R1	R1		
6			R4	R4	R4		
7			R3	R3	R3		

B) Construct the parsing table for the LR(1) parser.

The states of the LR(1) parser are the following:

- $I_0 = \{[S' \rightarrow \cdot S, \$], [S \rightarrow \cdot aSD, \$], [S \rightarrow \cdot b, \$]\}$
 - $I_1 = \text{goto}(I_0, S) = \{[S' \rightarrow S \cdot, \$]\}$
 - $I_2 = \text{goto}(I_0, a) = \{[S \rightarrow a \cdot SD, \$], [S \rightarrow \cdot aSD, e], [S \rightarrow \cdot b, e]\}$
 - $I_3 = \text{goto}(I_0, b) = \{[S \rightarrow b \cdot, \$]\}$
 - $I_4 = \text{goto}(I_2, S) = \{[S \rightarrow aS \cdot D, \$], [D \rightarrow \cdot Dc, \$], [D \rightarrow \cdot e, \$], [D \rightarrow \cdot Dc, c], [D \rightarrow \cdot e, c]\}$
 - $I_5 = \text{goto}(I_2, a) = \{[S \rightarrow a \cdot SD, e], [S \rightarrow \cdot aSD, e], [S \rightarrow \cdot b, e]\}$
 - $I_6 = \text{goto}(I_2, b) = \{[S \rightarrow b \cdot, e]\}$
 - $I_7 = \text{goto}(I_4, D) = \{[S \rightarrow aSD \cdot, \$], [D \rightarrow D \cdot c, c/\$]\}$
 - $I_8 = \text{goto}(I_4, e) = \{[D \rightarrow e \cdot, c/\$]\}$
 - $I_9 = \text{goto}(I_5, S) = \{[S \rightarrow aS \cdot D, e], [D \rightarrow \cdot Dc, e], [D \rightarrow \cdot e, e], [D \rightarrow \cdot Dc, c], [D \rightarrow \cdot e, c]\}$
- $\text{goto}(I_5, a) = I_5$
 $\text{goto}(I_5, b) = I_6$



$I_{10} = \text{goto}(I_7, c) = \{[D \rightarrow Dc \cdot, c/ \$]\}$
 $I_{11} = \text{goto}(I_9, D) = \{[S \rightarrow aSD \cdot, e], [D \rightarrow D \cdot c, e/c]\}$
 $I_{12} = \text{goto}(I_9, e) = \{[D \rightarrow e \cdot, e/c]\}$
 $I_{13} = \text{goto}(I_{11}, c) = \{[D \rightarrow Dc \cdot, e/c]\}$

STATES	ACTIONS					GOTO	
	a	b	c	e	\$	S	D
0	S2	S3				1	
1					ACCEPT		
2	S5	S6				4	
3					R2		
4				S8			7
5	S5	S6				9	
6				R2			
7			S10		R1		
8					R4		
9				S12			11
10			R3		R3		
11			S13	R1			
12			R4	R4			
13			R3	R3			

C) Construct the parsing table for the LALR(1) parser.

The states of the LR(1) parser that can not be differentiated by means of the look-ahead symbols are:

- States 2 and 5 = State 25
- States 3 and 6 = State 36
- States 4 and 9 = State 49
- States 7 and 11 = State 711
- States 8 and 12 = State 812
- States 10 and 13 = State 1013

The resulting parsing table for the LALR(1) after joining these states is:



STATES	ACTIONS					GOTO	
	a	b	c	e	\$	S	D
0	S25	S36				1	
1					ACCEPT		
25	S25	S36				49	
36				R2	R2		
49				S812			711
711			S1013	R1	R1		
812			R4	R4	R4		
1013			R3	R3	R3		

D) Extract conclusions about which methods can be used or not and show how the input string *aabece* would be analyzed using the correct ones.

The three methodologies can be used due to there are not conflicts in the corresponding parsing tables.

SLR(1)			LR(1)		
STACK	INPUT	ACTION	STACK	INPUT	ACTION
0	aabece\$	S2		aabece\$	S2
0a2	abece\$	S5	0a2	abece\$	S5
0a2a5	bece\$	S6	0a2a5	bece\$	S6
0a2a5b6	ece\$	R2	0a2a5b6	ece\$	R2
0a2a2S4	ece\$	S8	0a2a5S9	ece\$	S12
0a2a2S4e8	ce\$	R4	0a2a5S9e12	ce\$	R4
0a2a2S4D5	ce\$	S7	0a2a5S9D11	ce\$	S13
0a2a2S4D5c7	e\$	R3	0a2a5S9D11c13	e\$	R3
0a2a2S4D5	e\$	R1	0a2a5S9D11	e\$	R1
0a2S4	e\$	S6	0a2S4	e\$	S8
0a2S4e6	\$	R4	0a2S4e8	\$	R4
0a2S4D5	\$	R1	0a2S4D7	\$	R1
0S1	\$	ACCEPT	0S1	\$	ACCEPT

LALR(1)		
STACK	INPUT	ACTION
0	aabece\$	S25
0a(25)	abece\$	S25
0a(25)a(25)	bece\$	S36
0a(25)a(25)b(36)	ece\$	R2
0a(25)a(25)S(49)	ece\$	S812
0a(25)a(25)S(49)e(812)	ce\$	R4
0a(25)a(25)S(49)D(711)	ce\$	S1013
0a(25)a(25)S(49)D(711)c(1013)	e\$	R3
0a(25)a(25)S(49)D(711)	e\$	R1
0a(25)S(49)	e\$	S812
0a(25)S(49)e(812)	\$	R4
0a(25)S(49)D(711)	\$	R1
0S1	\$	ACCEPT

E) Complete the parsing table for the SLR(1) parser to provide the user with detailed information about the different errors that can be detected during the parsing process.

Possible information that the SLR(1) parser can provide to inform the user when a string cannot be parsed due to an error (empty cells in the parsing table) is shown following:

STATES	ACTIONS					GOTO	
	a	b	c	e	\$	S	D
0	S2	S3	Waiting a or b	Waiting a or b	Waiting a or b	1	
1	Waiting end of string	Waiting end of string	Waiting end of string	Waiting end of string	ACCEPT		
2	S2	S3	Waiting a or b	Waiting a or b	Waiting a or b	4	
3	Waiting e or end	Waiting e or end	Waiting e or end	R2	R2		
4	Waiting e	Waiting e	Waiting e	S6			5
5	Waiting c, e or end	Waiting c, e or end	S7	R1	R1		
6	Waiting c, e or end	Waiting c, e or end	R4	R4	R4		
7	Waiting c, e or end	Waiting c, e or end	R3	R3	R3		

