A2+TA2+TAA2

VIT UNIVERSITY

(Estd. u/s 3 of UGC Act 1956)

School of Computing Science & Engineering

Continuous Assessment Test -II

CSE2002-Theory of Computation and Compiler Design

Time: 1:30 Hrs **Max.Marks:50**

Answers ALL the questions

1. (a) Language L generates exactly the regular expressions with alphabet {0, 1}., give a CFG and the corresponding parse tree for the string

$(0 \cup (10)^{\bullet}1)^{\bullet}$ **(4marks)**

Note: (,) , ∪, * are terminal symbols

(b).Prove that $\mathbf{L = \{va^{k+1} \mid v \in \{a, b\}^*, |v| = k\}}$ is not context-free language. **(4 marks)**

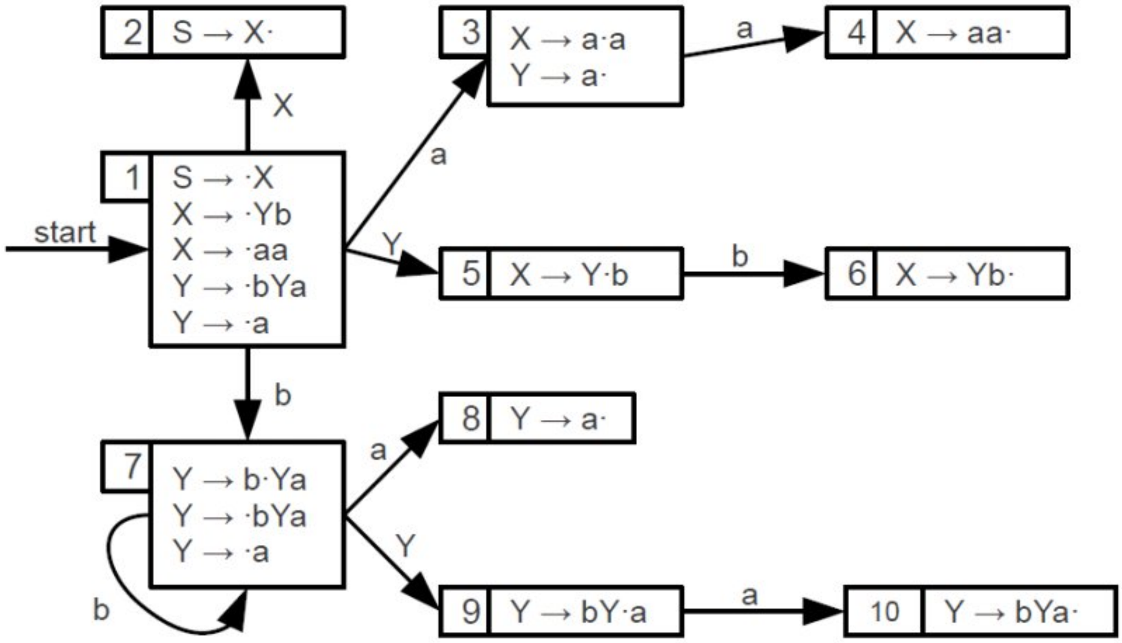
2(a) Consider the following LR(1) items in a single state of a shift/reduce parser. List any conflicts that exist and describe for what lookaheads they occur. **(4 marks)**

[A → a • bAb, d]
[B → a • bc, c]
[C → da • da, a]
[D → aa • d, c]
[E → ca • c, a]
[F → aAa • , a]
[G → ba • , a]
[H → Ha • , c]

(b). **(8 marks)**

S → X
X → Yb | aa
Y → a | bYa
Here is the associated LR(0) automaton for this grammar:

SEARCH VIT QUESTION PAPERS
ON TELEGRAM TO JOIN



i. Why is this grammar not SLR(1)?

ii. Using these FOLLOW sets and the LR(0) automaton, construct the LALR(1) lookaheads for each reduce item in the automaton. Show your results. Note that there are a total of 6 reduce items in the automaton.

iii. Is this grammar LALR(1)? Why or why not?

iv. Is this grammar LR(1)? Why or why not?

3. Suppose that we want to describe Javastyle class declarations like these using a grammar:

class Car extends Vehicle

public class JavaIsCrazy implements Factory, Builder, Listener

public final class President extends Person implements Official

One such grammar for this is

G={ (1) C → P F class identifier X Y

(2) P → public

(3) P → ε

(4) F → final

(5) F → ε

(6) X → extends identifier

(7) X → ε

(8) Y → implements I

(9) Y → ε

(10) I → identifier J

(11) J → , I **(note the comma before the I)**

(12) J → ε **}**

Your job is to construct an LL (1) parser table for this grammar. For reference, the terminals in this grammar are **public final class identifier extends implements , $**

Where **$** is the end of input marker, and the non-terminals are **C P F X Y I J**.

NOTE: **C** is a start symbol

(i).Compute the FIRST sets for each of the non-terminals in this grammar. Show your result.

(ii).Compute the FOLLOW sets for each of the non-terminal in this grammar. Show your result.

(iii).Using your results from (i) and (ii), construct the LL (1) parser table for this grammar. When indicating which productions should be performed, please use our numbering system from above. Show your result.

(iv) Using the above grammar, generate a string over terminals on your own and check the valid or invalid using the LL(1) table.

4. Give pushdown automata that recognize the following language.

$$\mathbf{L = \{a^n a^s d\ (ba)^s c^{2n} \mid n, s \geq 0\}}$$

Show that the PDA accepts the word **aaadbabacc.**

5. Construct a Turing machine that doubles each character in its input string. For example, if the input is 0100, then the machine should change its tape so it contains 00110000. Show all the states and transitions.

1(a). CFG:

$G = (V, \Sigma, R, S)$ with set of variables $V = \{S\}$, where S is the start variable; set of terminals $\Sigma = T$; and rules

$$S \to S \cup S \mid SS \mid S^* \mid (S) \mid 0 \mid 1 \mid \epsilon$$

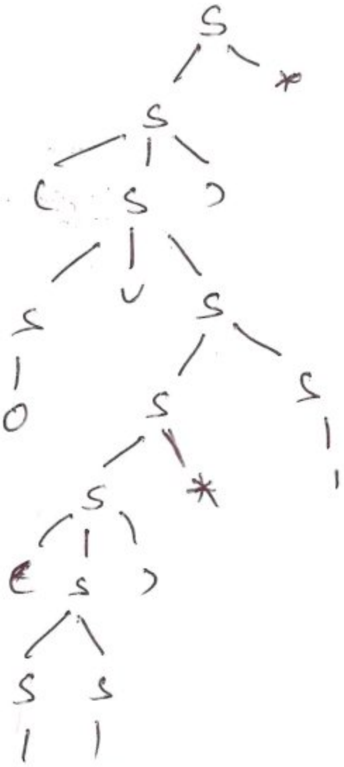
Derivation for $(0 \cup (10)^* 1)^*$ is

$$S \Rightarrow S^* \Rightarrow (S)^* \Rightarrow (S \cup S)^* \Rightarrow (0 \cup S)^* \Rightarrow (0 \cup SS)^*$$

$$\Rightarrow (0 \cup S^* S)^* \Rightarrow (0 \cup (S)^* S)^* \Rightarrow (0 \cup (SS)^* S)^*$$

$$\Rightarrow (0 \cup (1S)^* S)^* \Rightarrow (0 \cup (10)^* S)^* \Rightarrow (0 \cup (10)^* 1)^*$$

Parse tree:



1(b).

Since we can produce context free grammar, So the given language is CFL.

2(a).

Shift/reduce conflict:
[E →ca • c, a] and [H →Ha • , c] for lookahead c
Reduce/reduce conflict:
[F→aAa •, a] and [G → ba • , a] for lookahead a

2(b) (i) The FOLLOW set for Y contains **a** because of the production Y → bYa. Consequently, in state (3) we have a shift/reduce conflict, because on seeing an **a** we can't tell whether to shift it (from X → a·a) or to reduce it (because of Y → a·).

(ii) The updated lookahead sets are
LA (2, S → X·) = {**$**}
LA (3, Y → a·) = {**b**}
LA (4, X → aa·) = {**$**}
LA (6, X → Yb·) = {**$**}
LA (8, Y → a·) = {**a**}
LA (10, Y → bYa·) = {**a, b**}

(iii) The only way this grammar could not be LALR(1) is if we have a shift/reduce conflict in state 3, since it's the only state containing a reduce item and any other item. However, the lookahead here for Y → a· is **b**, which does not overlap with the shift item **a** as before. The grammar is thus LALR(1).

(iv) Since the grammar is LALR(1), it is also LR(1).

3.

(i) The FIRST sets are as follows:

FIRST(C) = {**public**, **final**, **class**}
FIRST(P) = {ε, **public**}
FIRST(F) = {ε, **final**}
FIRST(X) = {ε, **extends**}
FIRST(Y) = {ε, **implements**}
FIRST(I) = {**identifier**}
FIRST(J) = {ε, “**,**”}

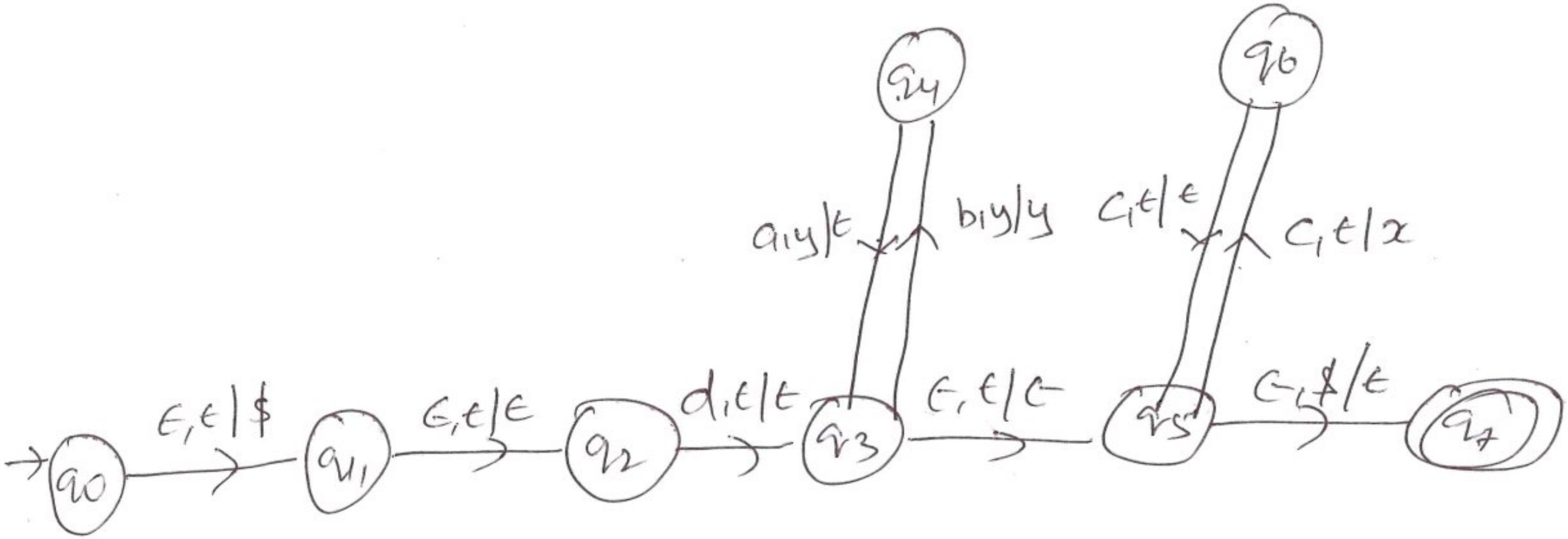
(ii) The FOLLOW sets are as follows:

FOLLOW(C) = {**$**}
FOLLOW(P) = {**final**, **class**}
FOLLOW(F) = {**class**}
FOLLOW(X) = {**implements**, **$**}
FOLLOW(Y) = {**$**}
FOLLOW(I) = {**$**}
FOLLOW(J) = {**$**}

(iii) The LL(1) parse table for the grammar is given below:

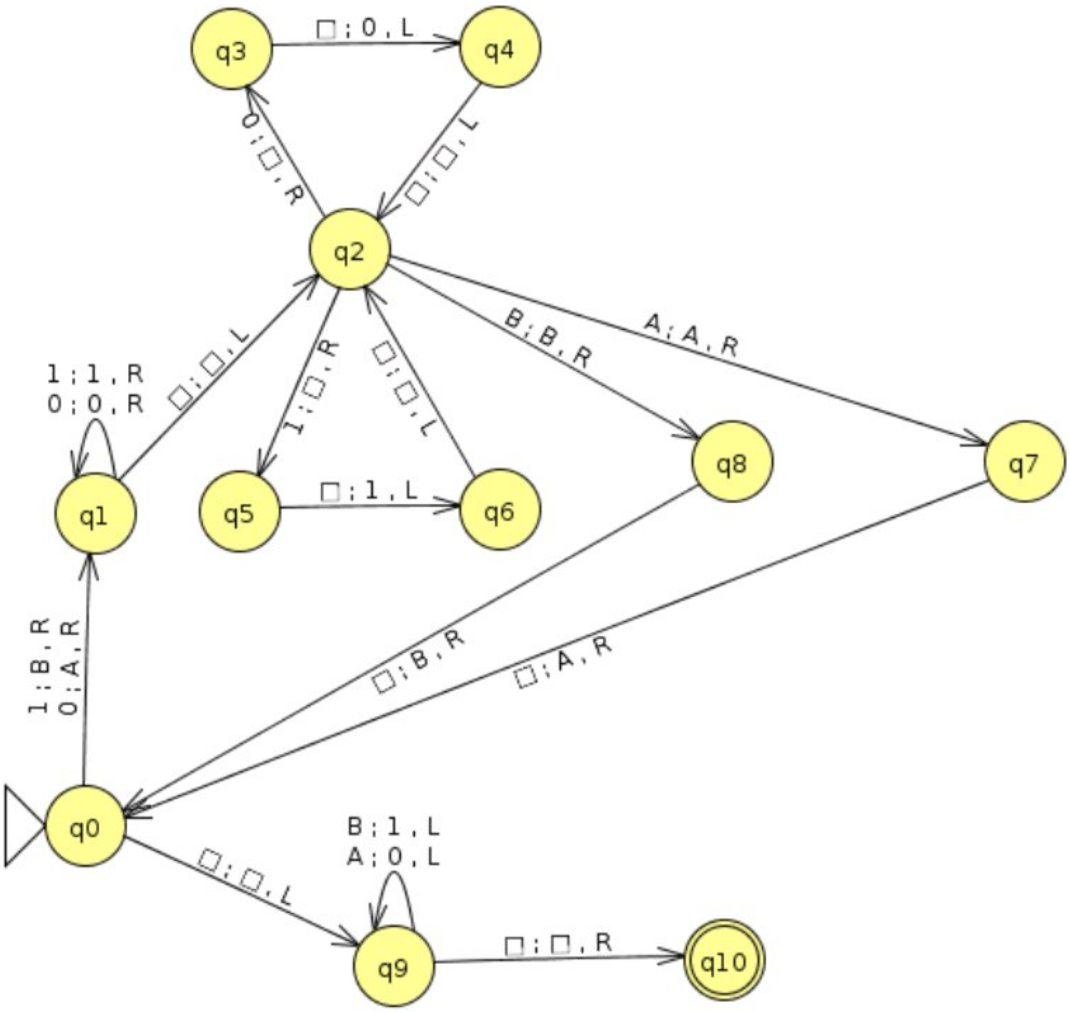
	public	**final**	**class**	**extends**	**implements**	**identifier**	**,**	**$**
C	1	1	1					
P	2	3	3					
F		4	5					
X				6	7			7
Y					8			9
I						10		
J							11	12

iv. Students can take any input string to show string is valid or not using parser table.



$(q_0, aaadbabacc, \epsilon) \vdash (q_1, aaadbabacc, \$) \vdash (q_1, aadbabacc, x\$) \vdash$
$(q_2, aadbabacc, x\$) \vdash (q_2, adbabacc, yx\$) \vdash (q_2, dbabacc, yyx\$) \vdash$
$(q_3, babacc, yyx\$) \vdash (q_4, abacc, yyx\$) \vdash (q_3, bacc, yx\$) \vdash$
$(q_4, acc, yx\$) \vdash (q_3, cc, x\$) \vdash (q_5, cc, x\$) \vdash (q_6, c, x\$)$
$\vdash (q_5, \epsilon, \$) \vdash (q_7, \epsilon, \epsilon)$

5.


q3
□ ; 0 , L
q4
0 ; □ , R
□ ; □ , L
q2
1 ; 1 , R
0 ; 0 , R
□ ; □ , L
1 ; □ , R
□ ; □ , L
B ; B , R
A ; A , R
q1
q5
□ ; 1 , L
q6
q8
q7
1 ; B , R
0 ; A , R
□ ; B , R
□ ; A , R
q0
B ; 1 , L
A ; 0 , L
□ ; □ , L
q9
□ ; □ , R
q10