Register number ______________________________

SRM Institute of Science and Technology
College of Engineering and Technology
School of Computing

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

Academic Year: 2022-23 (ODD)

B.Tech-Computer Science & Engineering

Test: CLA-T2 **Date: 19.10.2022**
Course Code & Title: 18CSC301T & Formal Languages and Automata Theory
Duration: 2 periods
Year & Sem: III Year /V Sem **Max. Marks: 50**

Set -A

Course articulation matrix:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO-1	3														3
CO-2		3	2												3
CO-3		3	3												3
CO-4		3	3												3
CO-5			3	1									2		3

Part - A

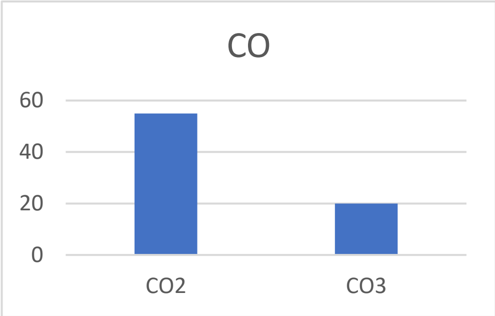
Instructions: Answer any two questions

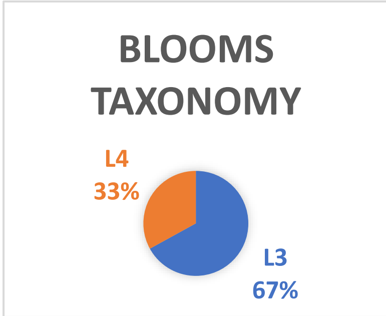
Q. No	Question	Marks	BL	CO	PO	PI Code
1	Consider the following grammar S → NP VP S → Aux NP VP S → VP NP → Det NOM NOM → Noun NOM → Noun NOM VP → Verb VP → Verb NP Det → that | this | a | the Noun → book | flight | meal | man Verb → book | include | read Aux → does i. How many productions in the given CFG are already in CNF? (1 Mark) a. 16 b. 12 c. 4 d. 13 ii. The given production are Type ______ grammar. (1 Mark) a. 0 b. 1 c. 2 d. 3 iii. List the terminal and non-terminal symbols (3 Marks)	25	3	2, 3	4	**4.2.1**

Register number ______________________________

	iv. Give the equivalent PDA rules for the grammar given in question (5 marks) v. Check if the above grammar could generate the string "does this flight include a meal" (4 marks) vi. Simplify the grammar (7 Marks) vii. Convert the above CFG to Chomsky Normal Form (CNF) (4 Marks)					
2	Read the following scenario and answer the following questions. Consider there are two color cubes (Red and Yellow) they are equal in number. The logic is Red cube to be taken and stack all the Red cubes first. Later once no more Red cubes are available, for each Yellow cube remove one Red cube from the stack. Make sure stack should be cleared. i. What is the maximum stack size for a PDA? (1 Mark) a. n b. 2 ^ n c. infinite d. n * n ii. Is the language generated for the given scenario is regular? (1 Mark) a. Yes b. No iii. Generate the accepting language for above Scenario. (3 Marks) iv. Construct CFG for the above Scenario. (4 Marks) v. Design PDA transitions for the given scenario. (5 marks) vi. List the PDA and CFG Tuple representations for above scenario. (4 Marks) vii. Illustrate a PDA Diagram for the above scenario. (4 Marks) viii. Check whether 3 consecutive yellow followed by three consecutive red balls can be taken? (3 Marks)	25	4	2, 3	4	**4.2.1**
3	Consider the following CFG for any programming construct BLOCK → STMT | { STMTS } STMTS → ε | STMT STMTS STMT → EXPR | if (EXPR) BLOCK | while (EXPR) BLOCK | do BLOCK while (EXPR) | BLOCK EXPR → a | constant | EXPR + EXPR | EXPR – EXPR | EXPR * EXPR| EXPR/EXPR i. What can be told about the given grammar? (1 Mark) a. It is ambiguous for the string a+a*a b. It is unambiguous for the string a+a*a c. It cannot derive the string a+a*a d. It can derive the string a+*a- ii. Which of the following is not true about ambiguous grammar? (1 Mark) a. It has two leftmost derivations. b. It has two rightmost derivations. c. It is sufficient to derive one leftmost and one rightmost derivation to prove its ambiguity. d. It has two parse trees. iii. Remove the null production (3 Marks) iv. Remove the unit production (4 Marks) v. Remove the useless symbols (4 Marks) vi. Convert it into GNF (12 Marks)	25	3	2	4	**4.2.1**

Register number ________________________________





SET A

(1) (i) ~~d~~ a

(ii) c

(iii) Terminals $\rightarrow$ {that, this, a, the, book, flight, meal, man, booth, include, read, does}.

NonTerminals $\rightarrow$ {S, NP, Nom, VP, Det, Noun, verb, Aux}.

(IV) PDA Rules

$$PDA = \{ \{q\}, \underbrace{\{\text{all the terminals}\}}_{\Sigma}, \delta, q, z_0, \Gamma, \phi \}$$

Rules

(1) $\delta(q, \epsilon, z_0) = (q, Sz_0)$

(2) For all NT,

$\delta(q, \epsilon, S) = \{ (q, NP\ VP), (q, Aux\ NP\ VP), (q, VP$

$\delta(q, \epsilon, NP) = (q, Det\ NOM)$

$\delta(q, \epsilon, Nom) = \{ (q, Noun), (q, Noun\ Nom$

$\delta(q, \epsilon, VP) = \{ (q, verb), (q, verb\ NP) \}$

$\delta(q, \varepsilon, \cancel{\text{Noun}}\ \text{Det}) = \{(q, \text{that}), (q, \text{this}), (q, a), (q, \text{the})\}$

$\delta(q, \varepsilon, \cancel{\text{Verb}}\ \text{Noun}) = \{(q, \text{book}), (q, \text{flight}), (q, \text{meal}), (q, \text{man})\}.$

$\delta(q, \varepsilon, \text{Aux}) = (q, \text{does}).$

$\delta(q, \varepsilon, \text{Verb}) = \{(q, \text{book}), (q, \text{include}), (q, \text{read})\}$

3) For all Terminals,

$\delta(q, \text{that}, \text{that}) = (q, \varepsilon)$

$\delta(q, \text{this}, \text{this}) = (q, \varepsilon)$

$\delta(q, a, a) = (q, \varepsilon)$

$\delta(q, \text{the}, \text{the}) = (q, \varepsilon)$

$\delta(q, \text{book}, \text{book}) = (q, \varepsilon)$

$\delta(q, \text{flight}, \text{flight}) = (q, \varepsilon)$

$\delta(q, \text{meal}, \text{meal}) = (q, \varepsilon).$

$\delta(q, \text{man}, \text{man}) = (q, \varepsilon)$

$\delta(q, \text{book}, \text{book}) = (q, \varepsilon)$

$\delta(q, \text{include}, \text{include}) = (q, \varepsilon)$

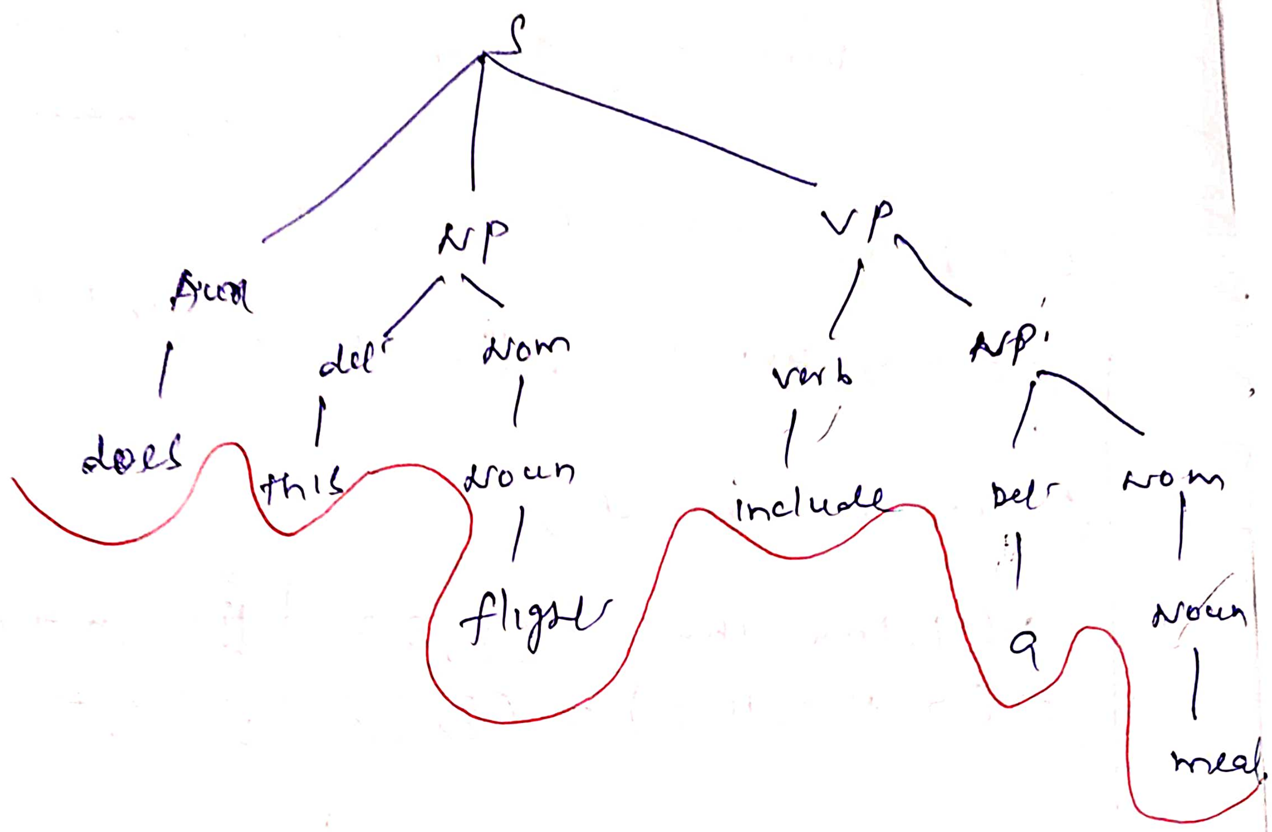
$\delta(q, \text{read}, \text{read}) = (q, \varepsilon)$

$\delta(q, \text{does}, \text{does}) = (q, \varepsilon)$

4) Final transition

$\delta(q, \varepsilon, z_0) = (q, \varepsilon)$

(v) w = "does this flight include a meal"



(vi) simplification

(i) No useless symbols.

(ii) No ε productions.

(iii) Elimination of UNIT prod.

ⓐ $S \rightarrow VP$
$\rightarrow$ verb
$\rightarrow$ book / include / read.

∴ $S \rightarrow$ book / include / read.

ⓑ. $VP \rightarrow$ verb
$\rightarrow$ book / include / read.

∴ $VP \rightarrow$ book / include / read.

(vii) CNF conversion

$S \rightarrow Aux \cdot NP \cdot VP$ not in CNF

$$\left.\boxed{\begin{array}{l} S \rightarrow Aux \cdot X \\ X \rightarrow NP\ VP \end{array}}\right\}$$ in CNF.

(2) (i) 0

(ii) not Regular (b)

(iii) $L = \{r^n y^n \mid n \geq 1\}$

(iv) $S(+)(A)$ $L = r^n y^n$

(u) equal no. of red & yellow.

$P = \boxed{\therefore S \rightarrow rSy \mid ry}$ Grammar (P1)

$G = (\{S\}, \{r, y\}, P, S)$

(v) PDA transition

For red cube $\rightarrow$ push into the stack

yellow cable $\rightarrow$ pop from the stack.

$\delta(q_0, r, z_0) = (q_0, r z_0)$

$\delta(q_0, r, r) = (q_0, rr)$

$\delta(q_0, y, r) = (q_1, \epsilon)$

$\delta(q_1, y, r) = (q_1, \epsilon)$

$\delta(q_1, \epsilon, z_0) = (q_2, \epsilon)$

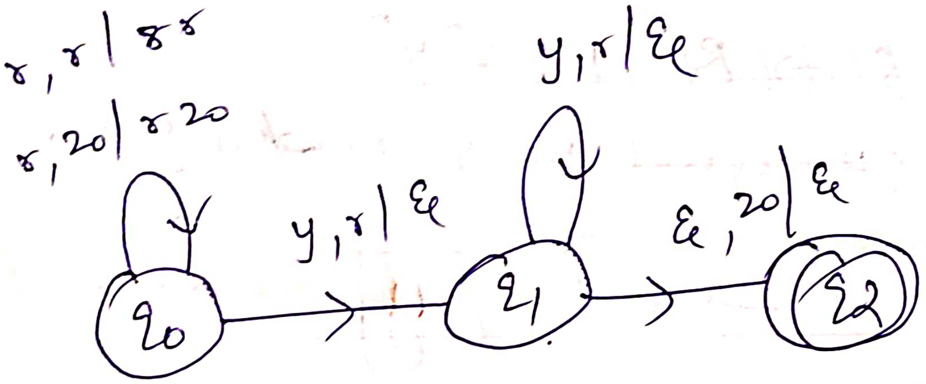
(VI)

$CFG = (\underset{N}{S}, \underset{T}{\{r, y\}}, \underset{P}{P}, \underset{S}{S})$

$PDA = (\{q_0, q_1, q_2\}, \{r, y\}, \underset{\Gamma}{\{r, z_0\}}, \delta, q_0, z_0, q_2, \cancel{F})$

3

(VII) <u>PDA Diagram</u>



(VIII) check for $r^3 y^3$

$\delta(q_0, rrryyy, z_0) \rightarrow (q_0, rryyy\epsilon, rz_0)$

$\rightarrow (q_0, ryyy\epsilon, rrz_0)$

$\rightarrow (q_0, yyy\epsilon, rrrz_0)$

$\rightarrow (q_1, yy\epsilon, rr z_0)$

$\rightarrow (q_1, y\epsilon, r z_0)$

$\rightarrow (q_1, \epsilon, z_0)$

$\rightarrow (q_2, \epsilon)$

$\hookrightarrow$ Accept

$(q_0, yyyrrr, z_0) \vdash (q_0,$

There is no transition for $\delta(q_0, y, z_0)$, so ~~the~~ ~~string~~ "yyyrrr" ~~is~~ 3 consecutive yellow followed by 3 consecutive red can not be taken.

(3). (i) a

(ii) c

(iii) Remove NULL production.

STMTS $\rightarrow \epsilon$ | STMT ; STMTS.

$\therefore$ NULL prod. is, $\boxed{STMTS \rightarrow \epsilon}$

BLOCK $\rightarrow$ STMT | {STMTS}

BLOCK $\rightarrow$ STMT | {STMTS} | { }.

STMTS $\rightarrow$ STMT. STMTS | STMT.

(IV) Remove ~~NULL~~ UNIT productions.

BLOCK → STMT → UNIT prod.

BLOCK → a | constant | EXPR + EXPR | EXPR − EXPR |
EXPR * EXPR | EXPR | EXPR | if (EXPR) BLOCK |
WHILE (EXPR) BLOCK | do BLOCK while (EXPR).

Stmts → STMT STMTS | a | constant ...

(V) Remove useless symbols.
No useless symbols.

Register number ______________________________

SRM Institute of Science and Technology
College of Engineering and Technology
School of Computing

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

Academic Year: 2022-23 (ODD)

B.Tech-Computer Science & Engineering

Test: CLA-T2 **Date: 19.10.2022**

Course Code & Title: 18CSC301T & Formal Languages and Automata Theory

Duration: 2 periods

Year & Sem: III Year /V Sem **Max. Marks: 50**

Set -B

Course articulation matrix:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO-1	3														3
CO-2		3	2												3
CO-3		3	3												3
CO-4		3	3												3
CO-5			3	1									2		3

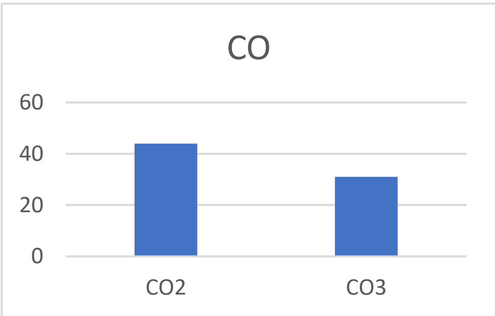
Part - A

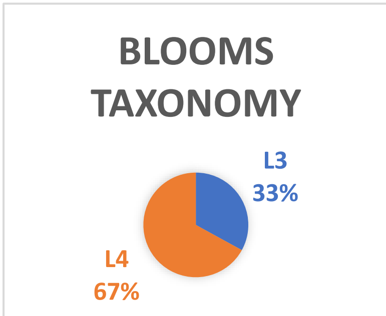
Instructions: Answer any two questions

Q. No	Question	Marks	BL	CO	PO	PI Code
1	Consider the following grammar S→ NP VP NP→DT N| NP PP PP→PRP NP VP→V NP| VP PP DT→a| the N→lion| deer| tree PRP→under |with |above V→ate| saw| ran i. Which of the following is not true about PDA? (1 Mark) a) PDA can be either deterministic or non-deterministic b) The stack in the PDA can have only stack symbols c) The stack in the PDA can contain stack symbols and input symbols d) For every CFG, there exists an equivalent PDA. ii. How many left recursive productions are there in the above grammar? (1 Mark)	25	3	2, 3	4	**4.2.1**

	a) 1 b) 2 c) no left recursive production d) 3 iii. Convert the above CFG into PDA. (5 Marks) iv. List the terminals and non-terminals in the given grammar (3 Marks) v. Can you derive the sentence " A lion ate the deer under the tree" using parse tree (3 Marks) vi. Simplify the grammar (8 Marks) vii. Convert the above CFG to Chomsky Normal Form (CNF) (4 Marks)					
2	Abinav has to travel back to his home every day from college. He can use two paths, path A and path B. Each month if he takes path A once he needs to take path B twice that month. In each given duration he needs to take path A first for n number of days sequentially followed by taking up path B as per the given condition. i. Which of the following is true about CFG? (1 Mark) a) The number of symbols in LHS of CFG must always be less than or equal to the number of symbols in RHS b) The RHS of the CFG cannot start with terminals c) The RHS of the CFG cannot start with nonterminals d) CFG cannot have epsilon in its RHS ii. I: Context sensitive grammar is a subset of Context Free Grammar (1 Mark) II: Regular grammars are the most restricted type of grammars a) Both are false b) Both are true c) I is false and II is true d) II is false and I is true iii. Construct Context Free Grammar (5 Marks) iv. Simplify the CFG (8 Marks) v. Convert the CFG into GNF (10 Marks)	25	4	2	4	**4.2.1**
3	The college organized a Teacher's day celebration event for all its employees. The employees participated in various games of the events. One such game is picking the color flowers from the pool. The employee has to pick the flowers in the order specified. The one who is picking all the flowers in the specified order at the earliest is the winner. The colored flowers are Red, Green, violet and yellow. **Case (i):** First, they should pick 'm' number of red flowers then 'n' number of green flowers then '4n' number of Violet flowers and at last '2m' number of yellow flowers. **Case (ii):** First they should pick 'n' number of red flowers then '3n' number of Green flowers. i. What can be inferred from the PDAs constructed for the given scenario? (1 Mark) a) The PDA constructed for Case i) is non deterministic b) The PDAs constructed for Case i) and Case ii) are deterministic	25	4	3	4	**4.2.1**

	c) The PDA constructed for Case ii) is non deterministic d) The PDAs constructed for Case i) and Case ii) are non deterministic ii. What can be said about the language accepted by a PDA with 12 stack elements? (1 Mark) a) Regular b) Context Free c) Recursive d) Nothing can be inferred iii. Write the Language for both cases. (5 marks) iv. Construction of PDA for both the cases. (8 marks) v. Illustrate a PDA Diagram for the above cases. (6 Marks) vi. Check whether 2 consecutive red flowers followed by 4 consecutive green flowers can be picked using ID in Case i)'s PDA ? (4 Marks)					





(1)

SRM Institute of Science and Technology

18CSC301T - Formal Language and Automata Theory

CT2 - Answer key - Set B

1. $S \to NP\ VP$

$NP \to DT\ N \mid NP\ PP$

$PP \to PRP\ NP$

$VP \to V\ NP \mid VP\ PP$

$DT \to a \mid the$

$N \to lion \mid deer \mid Tree$

$PRP \to under \mid with \mid above$

$V \to ate \mid saw \mid ran$

(i) c. The stack in the PDA can containg stack symbols and input symbols. (1)

(ii) b. 2 1 mark

5 marks

(iii) PDA = $\{\underbrace{\{q_0\}}_{Q}, \underbrace{\{a, the, lion, deer, Tree, under, with, above, ate, saw, ran\}}_{\Sigma}, \delta, q_0, Z_0, \Gamma, \underbrace{\phi}_{F}\}$

$\Gamma = \{Z_0, S, NP, PP, VP, DT, N, PRP, V\} \cup \Sigma$

δ

$\delta(q_0, \varepsilon, Z_0) = (q_0, S)$

Variables

$\delta(q_0, \varepsilon, S) = (q_0, NP\ VP)$

$\delta(q_0, \varepsilon, NP) = (q_0, DT\ N)$

$\delta(q_0, \varepsilon, NP) = (q_0, NP\ PP)$

$\delta(q_0, \varepsilon, PP) = (q_0, PRP\ NP)$

$\delta(q_0, \varepsilon, VP) = (q_0, V\ NP)$

$\delta(q_0, \varepsilon, VP) = (q_0, VP\ NP)$

$\delta(q_0, \varepsilon, DT) = (q_0, a)$

$\delta(q_0, \varepsilon, DT) = (q_0, the)$

$\delta(q_0, \varepsilon, PRP) = (q_0, under)$

$\delta(q_0, \varepsilon, PRP) = (q_0, with)$

$\delta(q_0, \varepsilon, PRP) = (q_0, above)$

$\delta(q_0, \varepsilon, V) = (q_0, ate)$

$= (q_0, saw)$

$= (q_0, ran)$

"

②

$\delta(q_0, \varepsilon, N) = (q_0, \text{lion})$

$\delta(q_0, \varepsilon, N) = (q_0, \text{deer})$

$\delta(q_0, \varepsilon, N) = (q_0, \text{tree})$

$\delta(q_0, a, a) = (q_0, \varepsilon)$

$\delta(q_0, \text{the}, \text{the}) = (q_0, \varepsilon)$

$\delta(q_0, \text{lion}, \text{lion}) = (q_0, \varepsilon)$

$\delta(q_0, \text{tree}, \text{tree}) = (q_0, \varepsilon)$

$\delta(q_0, \text{under}, \text{under}) = (q_0, \varepsilon)$

$\delta(q_0, \text{with}, \text{with}) = (q_0, \varepsilon)$

$\delta(q_0, \text{above}, \text{above}) = (q_0, \varepsilon)$

$\delta(q_0, \text{under}, \text{under}) = (q_0, \varepsilon)$

$\delta(q_0, \text{ate}, \text{ate}) = (q_0, \varepsilon)$

$\delta(q_0, \text{saw}, \text{saw}) = (q_0, \varepsilon)$

$\delta(q_0, \text{ran}, \text{ran}) = (q_0, \varepsilon)$

$\delta(q_0, \varepsilon, z_0) = (q_0, \varepsilon)$

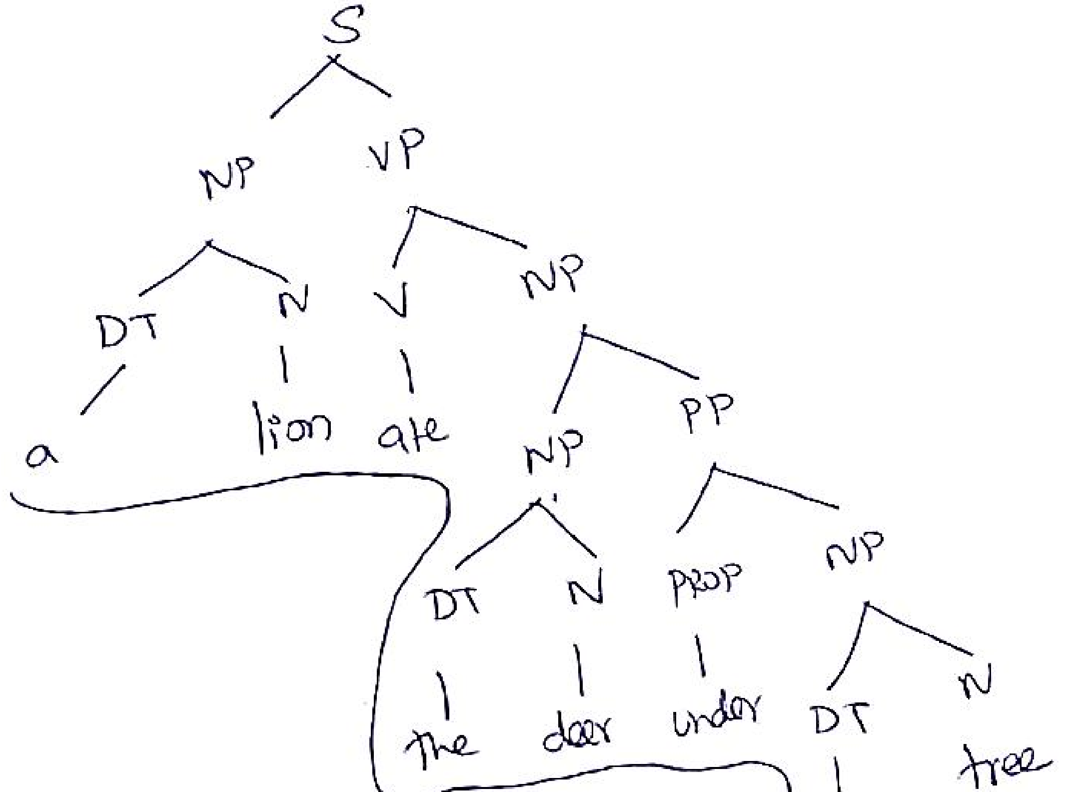
---

iv) Terminals = {a, the, lion, deer, tree, under, with, above, ate, saw, ran}

Non Terminals = {S, NP, PP, VP, DT, N, PRP, V}

-3marks

v)



vi) Simplification

i) Elimination of useless symbols

→ No non-generating and no not reachable symbols

ii) Elimination of ε production

→ No ε production

iii) Elimination of unit production

→ No unit production

vii) Chomsky Normal Form (CNF)

→ given grammar already in CNF form

---

2. $L = \{a^n b^{2n} \mid \Sigma = \{a, b\}^*\}$

i) a. The number of symbols in LHS of CFG must always be less than or equal to symbols in RHS.

ii) c. I is false and II is true

iii) Context Free Grammar:

$S \rightarrow aSbb \mid abb$

(or)

$S \rightarrow aAbb$

$A \rightarrow aAbb \mid \varepsilon$

IV) Simplification of grammar:

if grammar is

$S \to aSbb \mid abb$

1) Elimination of useless symbols
→ no non generating,
no - not reachable symbol

2) Elimination of ε production
→ No ε production

3) Elimination of unit production
→ no unit production

V) to GNF first grammar should be in CNF

CNF
$A \to a$ $B \to b$
$X_1 \to AS$ $X_2 \to BB$
$S \to X_1X_2 \mid AX_2 \mid \cancel{AX_2}$

GNF
$A \to a$ $B \to b$
$X_1 \to aS$ $X_2 \to bB$
$S \to aSX_2 \mid aX_2$

(or) if grammar is

$S \to aAbb$
$A \to aAbb \mid \varepsilon$

1) Elimination of useless symbols
→ no non-generating & no not reachable symbols

2) Elimination of ε production
$A \to \varepsilon$
$S \to aAbb \mid abb$
$A \to aAbb \mid abb$

3) Elimination of unit production
→ no unit production

V) to convert any grammar to GNF the grammar should be in CNF

CNF
$A \to a$ $B \to b$
$X_1 \to AS$ $X_2 \to BB$
$S \to X_1X_2 \mid AX_2$
$A \to X_1 X_2 \mid AX_2$

GNF
$A \to a$ $B \to b$
$X_1 \to aS$ $X_2 \to bB$
$S \to aSX_2 \mid aX_2$
$A \to aSX_2 \mid aX_2$

3. Case1: $R^m G^n V^{4n} Y^{2m}$ | Case2: $R^n G^{3n}$

i) b. The PDA's constructed for case 1 and case 2 are deterministic

ii) d. Nothing can be infered

iii) case 1:

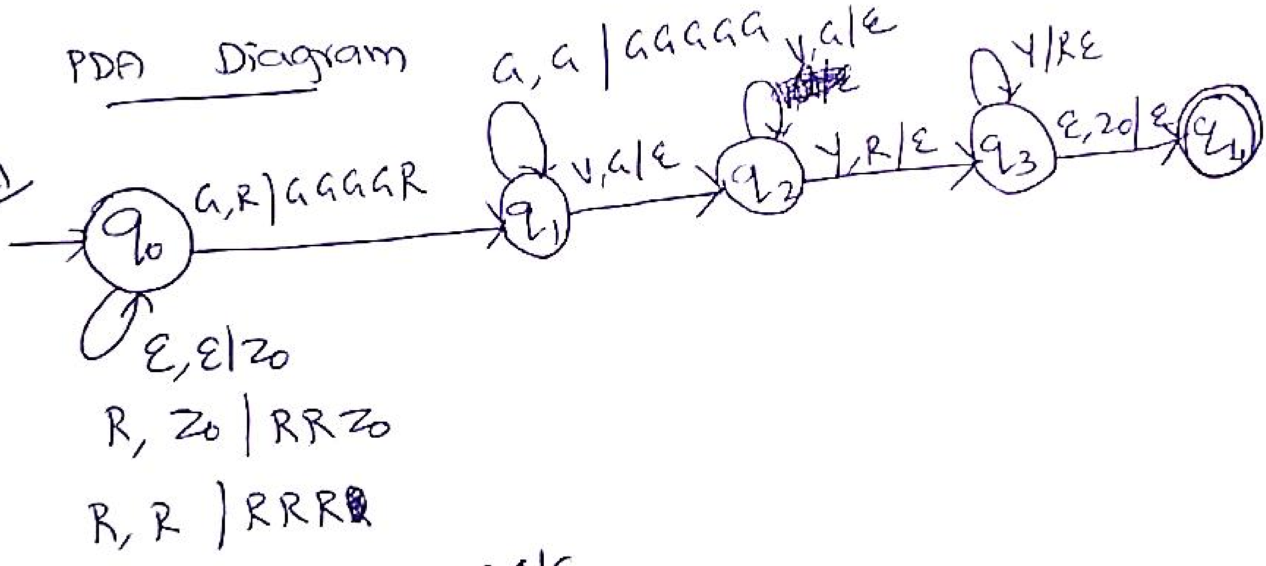
$L = \{R^m G^n V^{4n} Y^{2m} \mid \Sigma = \{R, G, V, Y\}\}$

case 2:

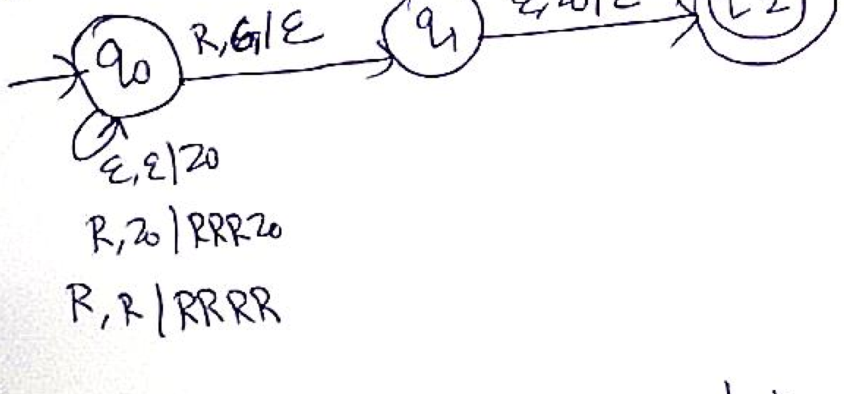
$L = \{R^n G^{3n} \mid \Sigma = \{R, G, V, Y\}\}$

v) PDA Diagram

Case1



Case2



vi) PDA for senarior → represent tuple notation for above

PDA = $\{Q, \Sigma, \delta, q_0, Z_0, \Gamma, F\}$

vi) W= RRGGGG

Case) ID

$(q_0, RRGGGG, \varepsilon)$

$\downarrow$

$(q_0, RRGGGG, Z_0)$

$\downarrow$

$(q_0, RGGGG, RRZ_0)$

$\downarrow$

$(q_0, GGGG, RRRRZ_0)$

$\downarrow$

$(q_0, GGG, GGGGRRRRZ_0)$

$\downarrow$

$(q_0, GG, GGGGGGGGRRRRZ_0)$

$\downarrow$

$(q_0, G, GGGGGGGGGGGGRRRRZ_0)$

$\downarrow$

$(q_0, \varepsilon, GGGGGGGGGGGGGGGGRRRRZ_0)$

$\downarrow$

at end of input transition not in final state and stack is not empty too.

Hence given i/p will be Rejected

———— x ————

Register number ______________________________

SRM Institute of Science and Technology
College of Engineering and Technology
School of Computing

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

Academic Year: 2022-23 (ODD)

B.Tech-Computer Science & Engineering

Test: CLA-T2 **Date: 19.10.2022**
Course Code & Title: 18CSC301T & Formal Languages and Automata Theory
Duration: 2 periods
Year & Sem: III Year /V Sem **Max. Marks: 50**

Set -C

Course articulation matrix:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO-1	3														3
CO-2		3	2												3
CO-3		3	3												3
CO-4		3	3												3
CO-5			3	1									2		3

Part - A

Instructions: Answer any two questions

Q. No	Question	Marks	BL	CO	PO	PI Code
1	Consider the grammar given below: S → NP VP S → Aux NP VP S → VP NP → Det NOM NOM → Noun NOM → Noun NOM VP → Verb VP → Verb NP Det → *that* | *this* | *a* | *the* Noun → *book* | *flight* | *meal* | *man* Verb → *book* | *include* | *read* Aux → *does* i) What can be told about the above grammar? (1 Mark) a) It has 3 unit productions b) It has 6 unit productions c) Certain terminals in the grammar cannot be derived d) Few useless symbols are inherently present in the grammar ii) I: Regular grammars are a subset of Context Free Grammars (1 Mark) II: Context free grammars are accepted by FSA a) I is true and II is false b) Both I and II are true c) II is true and I is false d) Both are false	25	3	2	4	**4.2.1**

	iii) List the terminals and non-terminals in the given grammar (3 Marks) iv) Check if the above grammar could generate the string "The flight include meal and the man read a book" (5 Marks) v) Simplify the grammar and then convert the above CFG to Chomsky Normal Form (CNF) (8+7 Marks)					
2	Consider the grammar given below which denotes Boolean expressions Expr → Expr or Term | Term Term → Term and Factor | Factor Factor → not Factor | (Expr) | true | false i) Which of the following lemma/ algorithm is not related to CFG? (1 Mark) a) Substitution rule b) Elimination of left recursion c) Pumping lemma for regular languages d) Elimination of useless symbols ii) A PDA can behave like a FSM when the stack size is____ (1 Mark) a) 1 b) 0 c)n d) infinite iii) Write leftmost derivation, rightmost derivation and parse tree for the string "true and not false or (false and (not true))" (6 Marks) iv) Is the grammar ambiguous? (2 Marks) v) Convert the given grammar to a Pushdown Automata (9 Marks) vi) Show any two reachable Instantaneous Descriptions (IDs) starting from the initial ID for the string (6 Marks)	25	3	2, 3	4	**4.2.1**
3	A school organized an annual day celebration for all students. The students participated in various games of the events. One of the game is picking the ball from the bag. The student has to pick the balls in the order specified. The one who is picking all the balls in the specified order at the earliest is the winner. The coloured balls are Red, Green, Violet, and Yellow.	25	4	2, 3	4	**4.2.1**

Register number ______________________________

Case (i): First, they should pick ‘n’ number of red balls then ‘m’ number of green balls then ‘m’ number of Violet balls and at last ‘n’ number of yellow balls.

Case (ii): First they should pick ‘n’ number of red balls then ‘m’ number of green balls then ‘m+n’ number of Violet balls.

i. Which of the following is not considered as an application of CFG? (1 Mark)

   a. Natural Language processing
   b. Syntax checking in programming languages
   c. Speech recognition
   d. Mathematical Induction

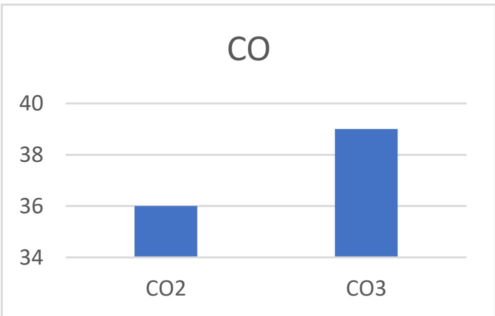
ii. A symbol that cannot derive a terminal or any reachable symbol from the starting symbol is called as_______ (1 Mark)

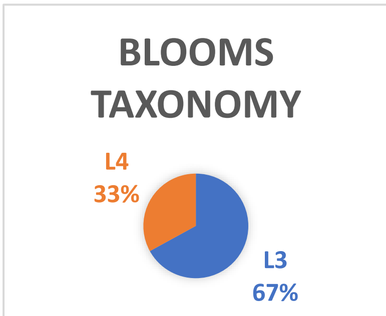
   a. Null symbol
   b. Reachable symbol
   c. Non reachable symbol
   d. Useless symbol

iii. Construction of PDA for the case (i) with transition diagram (8 Marks)

iv. Construction of PDA for the case (ii) with transition diagram (12 Marks)

v. Instantaneous Description for picking 2 red balls, 1green ball and 2 violet balls from PDA in case (ii) (3 Marks)





Register number ______________________________

1. Consider the grammar given below:

S → NP VP	Det → *that* | *this* | *a* | *the*
S → Aux NP VP	Noun → *book* | *flight* | *meal* | *man*
S → VP	Verb → *book* | *include* | *read*
NP → Det NOM	Aux → *does*
NOM → Noun	
NOM → Noun NOM	
VP → Verb	
VP → Verb NP	

i) What can be told about the above grammar? (1 Mark)
   a) It has 3 unit productions
   b) It has 6 unit productions
   c) Certain terminals in the grammar cannot be derived
   d) Few useless symbols are inherently present in the grammar

ii) I: Regular grammars are a subset of Context Free Grammars (1 Mark)
   II: Context free grammars are accepted by FSA
   a) I is true and II is false
   b) Both I and II are true
   c) II is true and I is false
   d) Both are false

iii) List the terminals and non-terminals in the given grammar (3 Marks)

iv) Check if the above grammar could generate the string "The flight include meal and the man read a book" (5 Marks)

v) Simplify the grammar and then convert the above CFG to Chomsky Normal Form (CNF) (8+7 Marks)

(i) a) It has 3 unit productions

(ii) a) I is true and II is false

(iii) Terminals → { that, this, a, the, book, flight, meal, man, include, read, does }

Non-Terminals → { S, NP, VP, Aux, Det, Nom, Verb, Noun }

(iv) string : " the flight include meal and the man read a book"

S ⇒ NP VP

S ⇒ Det NOM VP

S ⇒ the NOM VP

S ⇒ the Noun VP

S ⇒ the flight VP

S ⇒ the flight Verb NP

S ⇒ the flight include NP

S ⇒ the flight include det NOM

S ⇒ the flight include ?

So, from det, there is no possible transition which can give "meal" as string and also "and" is not present in any of the transition which is required for string to derive. Hence above grammar cannot generate given string.

(v) Simplification of Grammar

Step 1 : Eliminate ε production

There is no ε production.

Step 2 : Eliminate unit production

Unit Pairs	Productions
(S, S)	S → NP VP | Aux NP VP
(S, VP)	S → Verb NP
(S, verb)	S → book | include | read
(NP, NP)	NP → Det NOM
(NOM, NOM)	NOM → Noun NOM
(NOM, Noun)	NOM → book | flight | meal | man
(VP, VP)	VP → Verb NP

(VP, verb) VP → book | include | read

(det, det) det → that | this | a | the

(Noun, Noun) Noun → book | flight | meal | man

(Verb, Verb) Verb → book | include | read

(Aux, Aux) Aux → does

Now the grammar is,

S → NPVP | Aux NP VP | Verb NP | book

S → include | read

NP → Det NOM

NOM → Noun Nom | book | flight | meal | man

VP → Verb NP | book | include | read

det → that | this | a | the

Noun → book | flight | meal | man

Verb → book | include | read

Aux → does

Step3 : Eliminate useless symbols

All symbols are generating & reachable

Hence the final grammar is,

S → NP VP | Aux NP VP | Verb NP | book |

S → include | read

NP → Det NOM

NOM → Noun NOM | book | flight | meal | man

VP → Verb NP | book | include | read

Det → that | this | a | the

Noun → book | flight | meal | man

Verb → book | include | read

Aux → does.

Converstion of CFG to CNF

Step1 : Optmize the grammar

The grammar is in optimized form.

Step2 : Introduce Non-terminals.

Not required

Step 3 : Break Productions

S → NP VP | Aux P_1 | Verb NP | book

S → include | read

NP → Det NOM

NOM → Noun NOM | book | Flight | meal | man

VP → Verb NP | book | include | read

Det → that | this | a | the

Noun → book | flight | meal | man

Verb → book | include | read

Aux → does

PI → NP VP

2. Consider the grammar given below which denotes Boolean expressions

Expr → Expr or Term | Term

Term → Term and Factor | Factor

Factor → not Factor | (Expr) | true | false

i) Which of the following lemma/ algorithm is not related to CFG? (1 Mark)

a) Substitution rule

b) Elimination of left recursion

c) Pumping lemma for regular languages

d) Elimination of useless symbols

ii) A PDA can behave like a FSM when the stack size is____ (1 Mark)

a) 1 b) 0 c)n d) infinite

iii) Write leftmost derivation, rightmost derivation and parse tree for the string "true and not false or (false and (not true))" (6 Marks)

iv) Is the grammar ambiguous? (2 Marks)

v) Convert the given grammar to a Pushdown Automata (9 Marks)

vi) Show any two reachable Instantaneous Descriptions (IDs) starting from the initial ID for the string (6 Marks)

(i) c) Pumping Lemma for regular languages

(ii) b) 0

(iii) string :- true and not false or (false and (not true))

Left most derivation

Expr $\overset{lm}{\Rightarrow}$ Excpr or Term

$\Rightarrow$ Term or Term

$\Rightarrow$ Term and factor or Term

$\Rightarrow$ Factor and Factor or Term

$\Rightarrow$ true and Factor or Term

$\Rightarrow$ true and not Factor or Term

⇒ true and not false or Term

⇒ true and not false or Factor

⇒ true and not false or (Expr)

⇒ true and not false or (Term)

⇒ true and not false or
(Term and Factor)

⇒ true and not false or
(Factor and Factor)

⇒ true and not false or
(False and Factor)

⇒ true and not false or
(False and (Expr))

⇒ true and not False or
(False and (Term))

⇒ true and not false or (False
and (Factor))

⇒ true and not False or (False
and (not Factor))

⇒ true and not False or (False
and (not true))

Right most derivation

$Expr \overset{rm}{\Rightarrow} Expr$ or Term

$\Rightarrow$ Expr or Factor

$\Rightarrow$ Expr or (Expr)

$\Rightarrow$ Expr or (Term)

$\Rightarrow$ Expr or (Term and Factor)

$\Rightarrow$ Expr or (Term and (Expr))

$\Rightarrow$ Expr or (Term and (Term))

$\Rightarrow$ Expr or (Term and (Factor))

$\Rightarrow$ Expr or (Term and (not Factor))

$\Rightarrow$ Expr or (Term and (not true))

$\Rightarrow$ Expr or (Factor and (not true))

$\Rightarrow$ Expr or (false and (not true))

$\Rightarrow$ Term or (false and (not true))

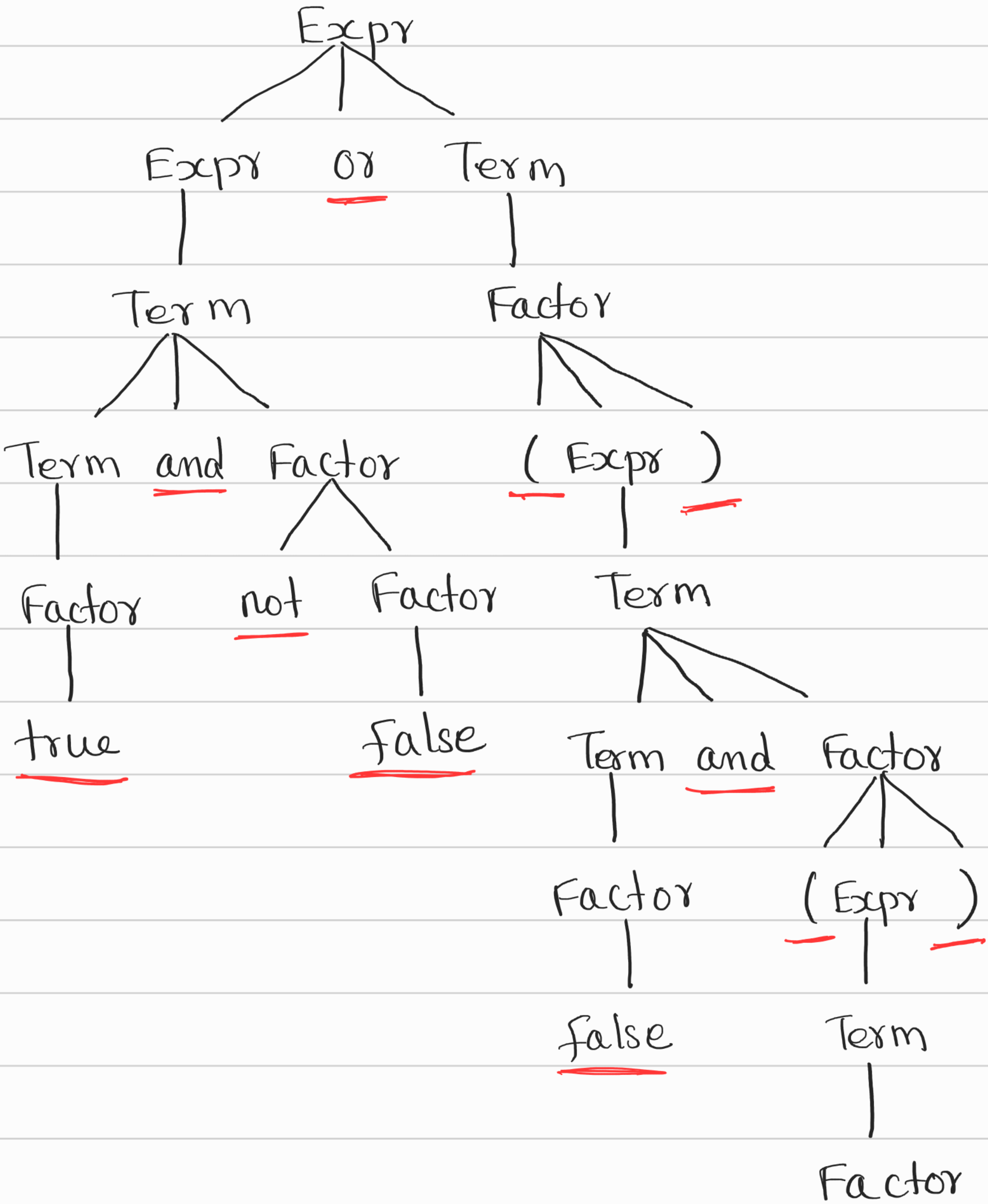
$\Rightarrow$ Term and Factor (false and (not true))

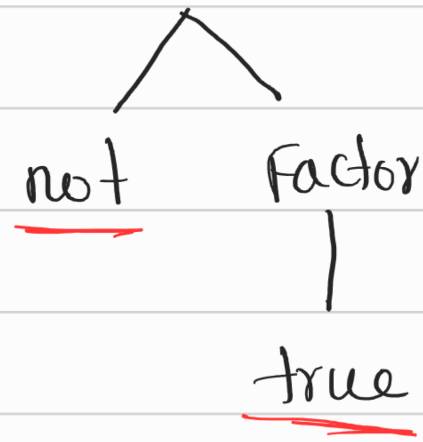
$\Rightarrow$ Term and not Factor (False and (not true))

$\Rightarrow$ Term and not false (false and (not true))

$\Rightarrow$ Factor and not false (false and (not true))

Parse Tree





(iv) Expr $\Rightarrow$ Term

$\Rightarrow$ Term and Factor

$\Rightarrow$ true and Factor

$\Rightarrow$ true and not Factor

We can't find the different derivation for the given string. So the grammer is not ambiguous

(v) Convert grammar to PDA

Expr $\rightarrow$ Expr or Term | Term

Term $\rightarrow$ Term and Factor | Factor

Factor $\rightarrow$ not Factor | (Expr) | true | false

The grammar can be converted to a PDA that accepts by empty stack as follows:

$$P = (\{q\}, \Sigma, \Gamma, \delta, q_0, Z_0)$$

That is,

$$P = (\{q\}, \{or, and, not, (,), true, false\}, \{or, and, not, (,), true, false, Expr, Term, Factor\}, \delta, q, S)$$

where δ is defined as,

$\delta(q, \varepsilon, Expr) = \{(Expr\ or\ Term), (Term)\}$

$\delta(q, \varepsilon, Term) = \{(Term\ and\ Factor), (Factor)\}$

$\delta(q, \varepsilon, Factor) = \{(not\ Factor), ((Expr)), (true), (false)\}$

$\delta(q, or, or) = \{(q, \varepsilon)\}$

$\delta(q, and, and) = \{(q, \varepsilon)\}$

$\delta(q, not, not) = \{(q, \varepsilon)\}$

$\delta(q, (, () = \{(q, \varepsilon)\}$

$\delta(q,),)) = \{(q, \varepsilon)\}$

$\delta(q, true, true) = \{(q, \varepsilon)\}$

$\delta(q, false, false) = \{(q, \varepsilon)\}$

(VI) String(i) :- true and not false (Any string can be taken)

Leftmost Derivation (Not required)

$$\begin{aligned}
Expr &\overset{lm}{\Longrightarrow} Term \\
&\Longrightarrow Term \text{ and } Factor \\
&\Longrightarrow Factor \text{ and } Factor \\
&\Longrightarrow true \text{ and } Factor \\
&\Longrightarrow true \text{ and not } Factor \\
&\Longrightarrow true \text{ and not } false
\end{aligned}$$

Simulation of the PDA

$(q, \text{true and not false}, Expr)$

$\vdash (q, \text{true and not false}, Term)$

$\vdash (q, \text{true and not false}, Term \text{ and } Factor)$

$\vdash (q, \text{true and not false}, Factor \text{ and } Factor)$

$\vdash (q, \text{true and not false}, \text{true and } Factor)$

$\vdash (q, \text{and not false}, \text{and } Factor)$

$\vdash (q, \text{not false}, Factor)$

$\vdash (q, \text{not false}, \text{not false})$

$\vdash (q, \text{false}, \text{false})$

$\vdash (q, \varepsilon, \varepsilon)$

String(ii): (false and (not true)) (Any string can be taken)

Left most Derivation (Not required)

$$\begin{aligned}
Expr &\overset{lm}{\Longrightarrow} Term \\
&\Longrightarrow Factor \\
&\Longrightarrow (Expr) \\
&\Longrightarrow (Term) \\
&\Longrightarrow (Term \text{ and } Factor) \\
&\Longrightarrow (Factor \text{ and } Factor) \\
&\Longrightarrow (\text{false and } Factor) \\
&\Longrightarrow (\text{false and } (expr)) \\
&\Longrightarrow (\text{false and } (term)) \\
&\Longrightarrow (\text{false and } (Factor)) \\
&\Longrightarrow (\text{false and } (\text{not } Factor)) \\
&\Longrightarrow (\text{false and } (\text{not true}))
\end{aligned}$$

Simulation of the PDA

$(q, \text{(false and (not true))}, Expr)$

$\vdash (q, \text{(false and (not true))}, Term)$

$\vdash (q, \text{(false and (not true))}, Factor)$

$\vdash (q, \text{(false and (not true))}, (Expr))$

$\vdash (q,\ \text{false and (not true))},\ \text{Expr}))$

$\vdash (q,\ \text{false and (not true))},\ \text{Term}))$

$\vdash (q,\ \text{false and (not true))},\ \text{Term and Factor}))$

$\vdash (q,\ \text{false and (not true))},\ \text{Factor and Factor}))$

$\vdash (q,\ \text{false and (not true))},\ \text{false and Factor}))$

$\vdash (q,\ \text{and (not true))},\ \text{and Factor}))$

$\vdash (q,\ \text{(not true))},\ \text{Factor}))$

$\vdash (q,\ \text{(not true))},\ \text{(Expr)}))$

$\vdash (q,\ \text{not true))},\ \text{Expr}))))$

$\vdash (q,\ \text{not true))},\ \text{Term}))))$

$\vdash (q,\ \text{not true))},\ \text{Factor}))))$

$\vdash (q,\ \text{not true))},\ \text{not Factor}))))$

$\vdash (q,\ \text{true))},\ \text{Factor}))))$

$\vdash (q,\ \text{true))},\ \text{true}))))$

$\vdash (q,\ )),\ ))))$

$\vdash (q,\ ),\ ))$

$\vdash (q,\ \varepsilon,\ \varepsilon)$

3. A school organized an annual day celebration for all students. The students participated in various games of the events. One of the game is picking the ball from the bag. The student has to pick the balls in the order specified. The one who is picking all the balls in the specified order at the earliest is the winner. The coloured balls are Red, Green, Violet, and Yellow.

Case (i): First, they should pick 'n' number of red balls then 'm' number of green balls then 'm' number of Violet balls and at last 'n' number of yellow balls.

Case (ii): First they should pick 'n' number of red balls then 'm' number of green balls then 'm+n' number of Violet balls.

i. Which of the following is not considered as an application of CFG? (1 Mark)

a. Natural Language processing
b. Syntax checking in programming languages
c. Speech recognition
d. Mathematical Induction

ii. A symbol that cannot derive a terminal or any reachable symbol from the starting symbol is called as_______ (1 Mark)

a. Null symbol
b. Reachable symbol
c. Non reachable symbol
d. Useless symbol

iii. Construction of PDA for the case (i) with transition diagram (8 Marks)
iv. Construction of PDA for the case (ii) with transition diagram (12 Marks)
v. Instantaneous Description for picking 2 red balls, 1green ball and 2 violet balls from PDA in case (ii) (3 Marks)

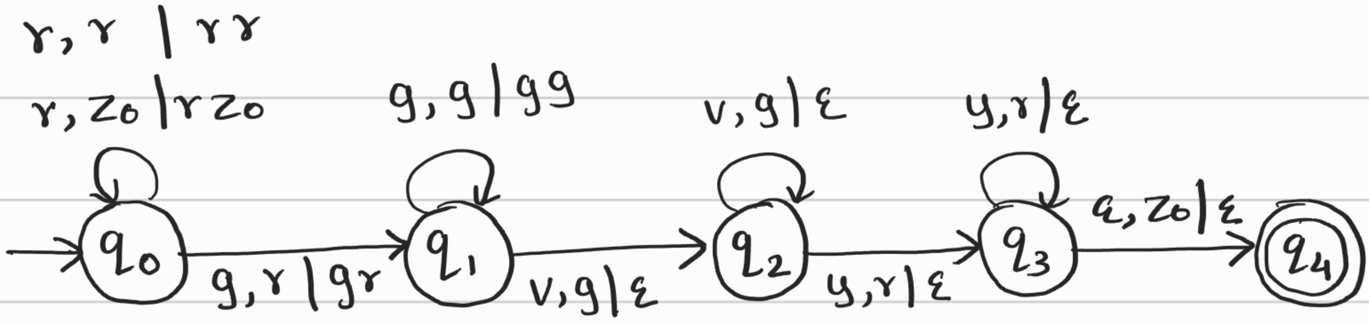
(i) d) Mathematical Induction

(ii) c) Non reachable Symbol

(iii) $L = \{ r^n g^m v^m y^n \}$

Logic for designing the PDA

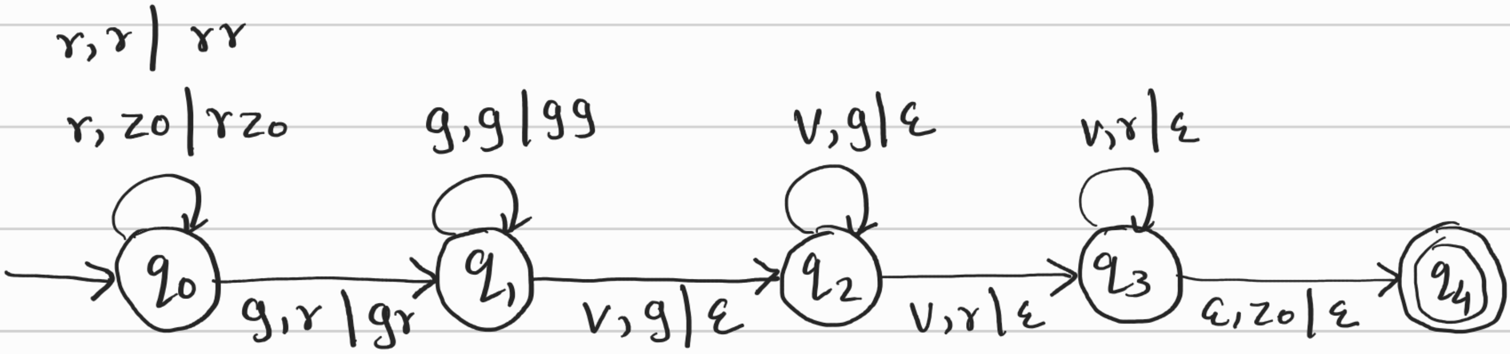
1. Push all r's into the stack.
2. Push all g's into the stack.
3. For every v we read, pop one g from the stack.
4. For every y we read, pop one r from the stack.
5. On consuming all the input if we reach the end of the stack, accept the input.



(iv) $L = \{ r^n g^m v^{m+n} \}$

Logic for designing the PDA

1. Push n r's into the stack.
2. Push m g's into the stack.
3. For m v's pop m g's and for n v's pop n r's from the stack.
4. When we finish reading the input, if the stack is empty, the input is accepted.



(v) $(q_0, rrgvv, Z_0) \vdash (q_0, rgvv, rZ_0)$

$\vdash (q_0, gvv, rrZ_0)$

$\vdash (q_1, vv, grrZ_0)$

$\vdash (q_2, v, rrZ_0)$

$\vdash (q_2, \varepsilon, \boxed{r}Z_0)$

The above string is not accepted because the stack not empty.

Register number ______________________________

SRM Institute of Science and Technology
College of Engineering and Technology
School of Computing

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

Academic Year: 2022-23 (ODD)

B.Tech-Computer Science & Engineering

Test: CLA-T2 **Date: 19.10.2022**

Course Code & Title: 18CSC301T & Formal Languages and Automata Theory

Duration: 2 periods

Year & Sem: III Year /V Sem **Max. Marks: 50**

Set -D

Course articulation matrix:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO-1	3														3
CO-2		3	2												3
CO-3		3	3												3
CO-4		3	3												3
CO-5			3	1									2		3

Part - A

Instructions: Answer any two questions

Q. No	Question	Marks	BL	CO	PO	PI Code
1	Consider the following grammar, S--> NP VP NP -->AD NO | DE NO | DE AD NO | NO VP -->VV NP DE -->DE DE | the | a | in | with AD --> red | short | tall | green VV --> lives | swings | walks NO --> girl | boy | game | dress | house (i) What is the use of Pumping lemma? (1 Mark) a) To prove the equivalence of NFA and DFA b) To prove the equivalence of ϵ-NFA and DFA c) To prove that the language is not regular d) To derive the equivalent regular expression from FSM ii) Which among the following CFG derives only alpha numeric characters? (1 Mark) a) S-->BA|AB|A|B; A-->0|1|..|9; B-->A|b|…|Z|a|b|…|z b) S-->BA|AB|A; A-->0|1|..|9; B-->A|b|…|Z|a|b|…|z c) S-->BA|AB|B; A-->0|1|..|9; B-->A|b|…|Z|a|b|…|z	25	3	2	4	**4.2.1**

Notation	Meaning
NP	Noun Phrase
VP	Verb Phrase
VV	Verb
NO	Noun
AD	Adjective
DE	Determiner

Register number ______________________________

<table>
<tr><td></td><td>d) S-->BA|AB; A-->0|1|..|9; B-->A|b|…|Z|a|b|…|z
iii) List the terminal and non-terminal symbols in the given grammar (3 Marks)
(iv) Derive the parse tree for the string “The short girl lives in the green house” (3 Marks)
(v) Check if the grammar could generate the string “The tall boy hided in the house” unambiguously? Justify your answer (5 Marks)
(vi) Convert the above CFG to Chomsky Normal Form (CNF) (12 Marks)</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>Consider the following grammar

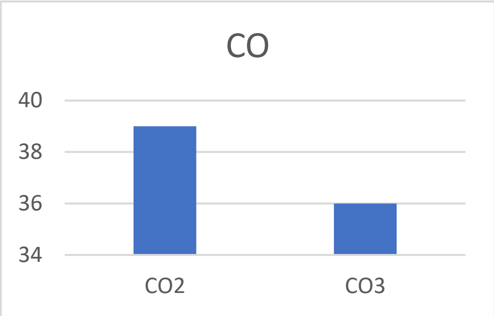
Stmt → if Cond then Stmt
Stmt → if Cond then Stmt else Stmt
Stmt → while Cond do Stmt
Stmt → id = Expr
Stmt → id
Cond → id Relop id
Expr → id Op id
Expr → id
Relop → < | > | <= |>= | == | !=
Op → + | - | * | / | %
id → a | b | c | d | e | f
i. Which of the given languages are accepted by Non Deterministic PDA but not by Deterministic PDA? (1 Mark)
a. Language generating strings that contain at least one symbol repeated at least twice
b. Even length Palindromes
c. Strings ending with a particular symbol
d. Strings starting with particular symbol
ii. Which of the following is not a simplification procedure of CFG? (1 Mark)
a. Elimination of null symbols
b. Elimination of unit productions
c. Elimination of generating symbols
d. Elimination of useless productions
iii. Construct leftmost derivation for “while a<b do c=d” (4 Marks)
iv. Construct rightmost derivation for “if a>b then c=d*e” (4 Marks)
v. Prove that the grammar is ambiguous grammar by using “if a<=b then if c>=d then e=f else f=e” (5 Marks)
vi. Convert the CFG into PDA. (7 Marks)
vii. Check whether the string “a=b/c” is accepted by PDA (3 Marks)</td><td>25</td><td>3</td><td>2, 3</td><td>4</td><td>4.2.1</td></tr>
</table>

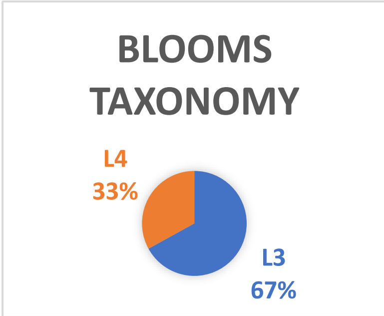
Register number ______________________________

<table>
<tr>
<td>3</td>
<td>The esteem institute conducts the placement for all the final year students. The students participated in various rounds of the placements. One such round is choosing the pattern of the written exam. The student has to choose the questions in the order specified. The one who is choosing all the questions in the specified order of answering at the earliest is the winner. The questions are MCQ, FILL IN THE BLANKS, MATCH THE FOLLOWING, DESCRIPTIVE.
Case (i)
First, they should choose and answer ‘n’ number of MCQ then ‘2n’ number of FILL IN THE BLANKS then ‘m’ number of MATCH THE FOLLOWING and at last ‘m’ number of DESCRIPTIVE.
Case (ii)
Or else First, they should choose and answer ‘n’ number of MCQ then ‘3n’ number of FILL IN THE BLANKS then ‘p’ number of MATCH THE FOLLOWING and at last ‘2p’ number of DESCRIPTIVE.
The order should not be changed. Design the suitable pushdown automata to check the order and satisfies the above cases. Illustrate with an example pattern.
i. The transition rule δ(q, b,Z)=(p,Z) represents ______________ operation. (1 Mark)
a) Push
b) Pop
c) No change
d) no input
ii. Consider let LangA describes the languages accepted by PDA by final state and LangB is the languages accepted by empty stack. Then (1 Mark)
a) Both languages are equal
b) LangA is subset of LangB
c) LangB is subset of LangA
d) Nothing can be inferred from both languages
iii. Construct a PDA with accepting state for the given statement Case i) (8 marks)</td>
<td>25</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4.2.1</td>
</tr>
</table>

Register number ________________________________

	iv. Construct a PDA with accepting state for the given statement Case ii) (8 Marks) v. Formal Definition of Constructed PDA (4 Marks) vi. Give the conversion of above constructed PDAs (case i) and ii)) to empty stack (3 Marks)					





Set - D

1) (i) 'c' (mark-1)

To prove that the language is not regular

(ii) 'd' (mark-1)

$S \to BA \mid AB$

$A \to 0 \mid 1 \mid \dots 9$

$B \to A \mid B \mid \dots \mid z \mid a \mid b \mid \dots z.$

(iii) (mark-3)

Terminals

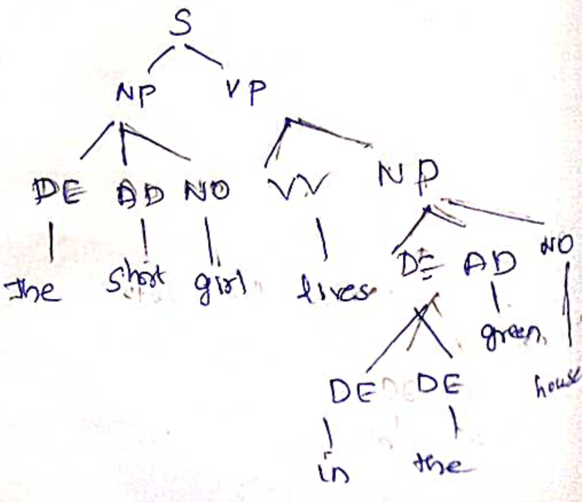
T = { the, a, in, with, red, short, tall, green, lives, swings, walks, girl, boy, game, dress, home }

Non terminals

V = { S, NP, VP, DE, AD, VV, NO }

(iv) Derive the parse tree for the ~~string~~ string. (mark 3)

"The short girl lives in the green house



S → NP VP

→ DE AD NO VP

→ The AD NO VP

→ The Short NO VP

→ The Short girl VP

→ The short girl VV NP

→ The short girl lives NP

→ The short girl lives DE AD NO

→ The short girl lives DE DE AD NO

→ The short girl lives in DE AD NO

→ The short girl lives in the AD NO

→ The short girl lives in the green NO

→ The short girl lives in the green house.

(mark-5)

(v) The given grammer cannot generate the string "The tall boy hided in the house" either ambigously or unambigously because "hided" ~~in the string~~ terminal is not found in grammer.

(mark-12)

(vi) Given CFG

S → NP VP

NP → AD NO | DE NO | DE AD NO | NO

VP → VV NP

DE → DE DE | the | a | is | win.

AD → red | short | tall | green.

VV → lives | swings | walk.

NO → girl | boy | game | dress |

house.

CNF

A grammer is said to be in chomsky normal form if it has production of form

NT → NT.NT

NT → T.

simplification of CFG before converting to CNF.

1> The given grammer does not contain any ~~word~~ useless symbol.

2> The given grammer does not contain 'ε' productions.

3> Elimination of unit productions.

Eleminating unit productions.

NP → NO we have.

S → NPVP

NP → AD NO | DE NO | DE AD NO

| girl | boy | game | dress |

house.

VP → VV NP

DE → DE DE | the | a | in | with

AD → red | short | tall | green.

VV → lives | swings | walks.

NO → girl | boy | game | dress | house.

Now, converting to CFG we have

S → NP VP

NP → AD NO

NP → DE NO

NP → DE NP [replacing AD NO → NP]

NP → girl | boy | game | dress | house.

VP → VP NP

DE → DE DE

DE → the | a | in | with.

AD → red | short | tall | green

VV → lives | swings | walks

NO → girl | boy | game | dress | house.

In the above grammar all the productions are of the form

NT → NT · NT

NT → T.

Thus converted to CNF.

2) (i) 'b' (Mark-1)

Even length palindromes

(ii) 'c' (Mark-1)

Elimination of generating symbols.

(iii) LMD of "while a<b do c=d" (4-marks)

$\beta\alpha$ = "while a<b do c=d"

stmt $\xrightarrow{lmd}$ while cond do stmt

$\xrightarrow{lmd}$ while id Relop id do stmt

→ while a Relop id do stmt

→ while a < id do stmt

→ while a < b do stmt

→ while a < b do id = Expr.

→ while a < b do c = Expr

→ while a < b do c = id

→ while a < b do c = d.

→ while a < b

(iv) RMD of "If a > b then c = d * e"
(4 marks)

Stmt $\xrightarrow{Rmd}$ If cond then stmt

→ If cond then id = Expr

→ If cond then id = id op id

→ If condn then id = id op e

→ If cond then id = id * e

→ If cond then id = d * e

→ If cond then c = d * e

→ If id relop id then c = d * e

→ If id relop b then c = d * e

→ If id > b then c = d * e

→ If a > b then c = d * e.

Thus rmd is achieved.

(V) (Mark - 5)

PT. the grammar is ambiguous grammar

using string
If a <= b then If c >= d then e = f
else f = e

Defn
If a grammar is having 2 lmd or 2 rmd then that grammar is ambiguous grammar.

I lmd

Stmt $\xrightarrow{lmd}$ If condn then Stmt else Stmt.

→ If id relop id then stmt else stmt

→ If a relop id then stmt else stmt

→ If a <= id then stmt else stmt.

→ If a <= b then stmt else stmt.

→ If a <= b then If cond then stmt
else stmt.

→ If a <= b then if id relop id then stmt
else stmt.

→ if a <= b then if a relop id then stmt
else stmt.

→ if a <= b then if c >= id then
stmt else stmt

→ if a <= b then if c >= d then stmt
else stmt

→ if a<=b then if c >= d then id = expr
else stmt

→ if a <= b then if c >= d then e = expr
else stmt

→ if a<=b then if c>=d then e = id
else stmt.

→ if a<=b then if c>=d then e = f
else stmt.

→ if a<=b then if c>=d then e = f
else id = expr

→ if a <= b then if c >= d then e = f
else f = expr

→ if a <= b then if c >= d then e = f
else f = id.

→ if a <= b then if c >= d then e = f
else f = e.

The string is derived using lmd → ①.

II - lmd

stmt $\xrightarrow{lmd}$ if condn then stmt.

→ if id relop id then stmt

→ if a relop id then stmt

→ if a <= id then stmt

→ if a <= b then stmt.

→ If a<=b then If condn then stmt else stmt

→ If a<=b then if id relop id then stmt else stmt.

→ If a<=b then if c relop id then stmt else stmt.

→ if a<=b then if c >= id then stmt else stmt.

→ If a<=b then if c>=d then stmt else stmt.

→ If a<=b then if c>=d then id = Expr else stmt

→ if a<=b then if c>=d then e = expr else stmt.

→ If a<=b then if c>=d then e = id else stmt

→ If a<=b then if c>=d then e = f else stmt.

→ If a<=b then if c>=d then e = f else id = Expr

→ If a<=b then if c>=d then e=f else f = expr

→ If a<=b then if c>=d then e=f else f = id

→ If a<=b then if c>=d then e=f else f=e.

The same string is derived using II lmd. The above grammer is having 2 lmd for the given string so it is ambiguous grammer.

(vi) CFG to PDA. (7 marks)

~~Stmt → if condn~~

PDA P that accepts L(G) by empty stack is as follows.

1\ ~~Pro~~ $P = (\{q\}, T, V \cup T, \delta, q, S)$

where δ is defined as:-

1> For each variable A_i

$$\delta(q, \epsilon, A) = \{(q, \beta)\}$$

where $A \to \beta$ is a production

2> For each nonterminal a

$$\delta(q, a, a) = \{(q, \epsilon)\}$$

Stmt → if condn then stmt |
if condn then stmt else stmt |
while condn do stmt |
id = Expr | id.

condn → id relop id

Expr → id op id | id

Relop → < | > | <= | >= | == | !=

op → + | - | * | / | %.

id → a | b | f | e | f.

PDA.

$\delta(q, \epsilon, stmt) = \{(q, \text{if condn then stmt}),$

$(q, \text{if condn then stmt else stmt}),$

$(q, \text{while condn do stmt}),$

$(q, id = Expr), (q, id)\}$

$\delta(q, \epsilon, condn) = \{(q, \text{id relop id})\}$

$\delta(q, \epsilon, Expr) = \{(q, \text{id relop id}), (q, id)\}$

$\delta(q, \epsilon, Relop) = ((q, <), (q, >), (q, >=)$

$(q, <=), (q, ==) (q, !=)\}$

$\delta(q, \epsilon, op) = \{(q, +), (q, -), (q, /), (q, \%)$

$(q, *)\}$

$\delta(q, id) = \{(q, a) (q, b) (q, c) (q, d)$

$(q, e) (q, f)\}$

$\delta(q, <, <) = (q, \epsilon)$

$\delta(q, >, >) = (q, \epsilon)$

$\delta(q, >=, >=) = (q, \epsilon)$

$\delta(q, <= <=) = (q, \epsilon)$

$\delta(q, ==, ==) = (q, \epsilon)$

$\delta(q, !=, !=) = (q, \epsilon)$

$\delta(q, +, +) = (q, \epsilon)$

$\delta(q, --) = (q, \epsilon)$

$\delta(q, *, *) = (q, \epsilon)$

$\delta(q, /, /) = (q, \epsilon)$

$\delta(q, 1, 1) = (q, \epsilon)$

$\delta(q, a, a) = (q, \epsilon)$

$\delta(q, b, b) = (q, \epsilon)$

$\delta(q, c, c) = (q, \epsilon)$

$\delta(q, d, d) = (q, \epsilon)$

$\delta(q, e, e) = (q, \epsilon)$

$\delta(q, f, f) = (q, \epsilon)$

$\delta(q, \epsilon, \epsilon) = (q, \epsilon)$ // accepted empty state

(viii)

A string a accepted by a PDA if it has a path from initial to final state

Eg, ~~E,~~ a=b|c

stmt → id = Expr

⇒ a = Expr

→ a = id op id

→ a = b op id

→ a = b|id

→ a = b|c.

State	i/p	stack
q_0	a = b | c	Empty
q_0	a = b | c	stmt
q_0	a = b | c	id = Expr
q_0	a = b | c	a = Expr
q_0	= b | c	= Expr
q_0	b | c	Expr
q_0	b | c	id op id
q_0	b | c	b op id
q	| c	op id
q	| c	| id
q	c	id
q	c	c
q	ϵ	ϵ // accepted empty stack

The above string is accepted by PDA.

3> (i) No change (1 Mark)

(ii) Both ~~lang~~ languages are equal (1 Mark)

(iii) Let x = MCQ

y = Fill up the blanks

z = Match the following

k = Descriptive.

Case (i) student should choose

$x^n y^{2n} z^m k^m$.

We can design a PDA by pushing all x and checking for every x there are $2y$'s and then push z to check its equality with k.

	Q	I/P	Stack	Q x operations.
1)	q_0	x	Z_0	(q_0, xZ_0)
2)	q_0	x	x	(q_0, xx)
3)	q_0	y	x	(q_1, ϵ)
4)	q_1	y	x	(q_2, ϵ)
5)	q_2	y	x	(q_1, x)
6)	q_2	z	Z_0	(q_3, zZ_0)
7)	q_3	z	z	(q_3, zz)
8)	q_3	k	z	(q_4, ϵ)
9)	q_4	k	Z_0	(q_4, ϵ)
10)	q_4	ϵ	Z_0	(q_5, Z_0)

Accept state.

(iv)

Case(ii) Student should choose

$x^n y^{3n} z^p k^{2p}$.

we can design a PDA by pushing all x and checking for every x there are 3 y's and the push all z and check for every z there are 2 k's.

	Q	i/p	Stack	Q x operations
			Z_0	(q_0, xZ_0)
1)	q_0	x	x	(q_0, xx)
2)	q_0	x	x	$(q_1, \cancel{x})$
3)	q_0	y	x	(q_2, x)
4)	q_1	y	x	(q_3, ε) (q_1, x)
5)	q_2	y	x	(q_4, zZ_0)
6)	q_3,	y	Z_0	
7)	q_3	z	z	(q_4, zz)
8)	q_4	z	z	$(q_5, \cancel{z})$
9)	q_4	k	z	(q_6, ε) (q_5, z)
10)	q_5	k	z	(q_7, Z_0)
	q_6	ϵ	Z_0	
11)	q_6			accepting stat,

(V) PDA

$P(Q, \Sigma, q_0, \delta, \Gamma, z_0\ F)$

Q = set of states.

Σ = set of i/p symbols

q_0 - initial state.

δ : transistion function

Γ : stack symbols

z_0 - bottom of stack

F - final state.

$P(\{q_0, q_1, q_2\}, \{x, y, z, k\}, q_0, \delta,$
$\{\cancel{x, y, z, k}\ x, z\}, z_0, \{q_2\})$

↳ Formal defn of constructed PDA.

(VI) Empty stack PDA.

(i) iv → problem replace ~~line~~ transistions 10 with

$$\delta(q_4, \epsilon, z_0) = (q_5, \epsilon)$$

// (empty stack)

(ii) (v) → problem replace transistions ~~10~~ 11 with

$$\delta(q_6, \epsilon, z_0) = (q_7, \epsilon)$$

// empty stack.