Total No. of Questions : 8]

SEAT No. :

PB3778

[Total No. of Pages :4

[6262]-36
T.E. (Computer Engineering)
THEORY OF COMPUTATION
(2019 Pattern) (Semester-I) (310242)

Time : 2½ Hours] *[Max. Marks : 70*

Instructions to the candidates:

1) *Answer Q.1 or Q.2, Q.3 or Q.4, Q.5 or Q.6, Q.7 or Q.8.*
2) *Neat diagrams must be drawn wherever necessary.*
3) *Figures to the right indicates full marks.*
4) *Assume suitable data, if necessary.*

Q1) a) What is context free Grammar? Define CFG. What are the capabilities of CFG? **[8]**

Give a context Free Grammar for the following language

$L = \{ w \in \{a,b\}^* \mid w \text{ is a palindrome of odd length} \}$.

b) i) What is Derivation in CFG?

ii) What is relation of parse tree for derivation in CFG?

iii) What is leftmost derivation and Rightmost derivation?

iv) Explain leftmost derivation and Rightmost derivation with parse tree. Derive the string a-b+c using leftmost derivation and Rightmost derivation for the CFG having production rule.

G = {S = S + S

S = S -S

S = a | b | c

} **[10]**

OR

P.T.O.

Q2) a) When do we say that CFG is in Greibach Normal Form (GNF)? Explain the steps to convert CFG to GNF for following Grammars **[12]**

G1 = {S → aAB | aB, A → aA | a, B → bB | b}

G2 = {S → aAB | aB, A → aA | ε, B → bB | ε}

G3 = {S → XB | AA

A → a | SA

B → b

X → a }

b) i) What is ambiguity in CFG? What is relation of parse tree for finding ambiguity in CFG.

ii) What is leftmost derivation and Rightmost derivation?

iii) Explain leftmost derivation and Rightmost derivation and ambiguity for the CFG having production rule.

G = { S = aSb | SS

S = ∈ }

[6]

Q3) a) What is pushdown automata? Define PDA pictorially and mathematically with respect to input tape, stack, finite control and Instanteous description.

Design a PDA for accepting a language $\{a^n b^{2n} | n>=1\}$ **[8]**

b) Construct a context free grammar which accepts N (A), where **[10]**

A = ({q0, q1}, {0, 1}, {Z0, Z}, δ ,q0, Z0, φ where δ is given by

δ (q0, 1, Z0) = {(q0, Z Z0)}

δ (q0, ε , Z0) = {(q0, ε)}

δ (q0, 1, Z) = {(q0, Z Z)}

δ (q0, 0, Z) = {(q1, Z)}

δ (q1, 1, Z) = {(q1, ε)}

δ (q1, 0, Z0) = {(q0, Z0)}

OR

Q4) a) Design a PDA for accepting a language $\{0^n1^m0^n \mid m, n>=1\}$. **[6]**

b) Draw a PDA for the CFG given below: **[6]**

$S \rightarrow aSb$

$S \rightarrow a \mid b \mid \varepsilon$

And simulate PDA to recognize "aaabb".

c) Design a push down automation to recognize the language generated by the following **[6]**

grammar?

$S \rightarrow S + S \mid S * S \mid 4 \mid 2$

Show the acceptance of the input string 2 + 2*4 by this PDA.

Q5) a) Elaborate the following terms with proper examples **[4]**

i) Universal Turing Machine (UTM)

ii) Recursively Enumerable Languages

b) Design a TM that multiplies two unary numbers over $\Sigma = \{1\}$. Write simulation for the string 11*111. **[7]**

c) Construct a TM for the language $L = \{0^n1^n2^n\}$ where $n \geq 1$. **[6]**

OR

Q6) a) Construct a TM for substraction of two unary numbers f(a-b) = c where a is always greater than b. **[5]**

b) What is undecidability? How do we prove universal language is undecidable? What is the relation between undecidability and reducibility theory. **[12]**

Q7) a) What do you mean by polynomial time reduction? Explain with an example of SAT. **[7]**

b) Explain the following terms with respect to computations complexity with example. **[10]**

i) Solvable Vs Unsolvable problem

ii) Decidable Vs. Undecidable problem

iii) P Vs NP problem

OR

Q8) a) Explain in brief the term "recursively enumerable". **[6]**

b) Explain examples of problems in NP. **[6]**

c) Differentiate between P class and NP class. **[5]**

ଌ ଌ ଌ