M.TECH/CSE/2ND SEM/CSEN 5234/2019
THEORY OF COMPUTATION
(CSEN 5234)

Time Allotted : 3 hrs **Full Marks : 70**

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and <u>any 5 (five)</u> from Group B to E, taking <u>at least one</u> from each group.

Candidates are required to give answer in their own words as far as practicable.

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternative for the following: **10 × 1 = 10**

(i) Consider the language L = $\{0^n1^{n+1}0^{n+2}1^{n+3} \mid n \geq 1\}$ on the alphabet{0 ,1 }. Which of the following alternatives is true?
(a) L can be recognized by a deterministic finite state acceptor (dfsa).
(b) L can be recognized by a deterministic pushdown acceptor (dpda) but not by a dfsa.
(c) L can be recognized by a non-deterministic pushdown acceptor (ndpda) but not by a dpda.
(d) L can be recognized by a Turing Machine (TM) but not by an ndpda.

(ii) Consider the grammar
$S \rightarrow aSa \mid bSb \mid a \mid b \mid \in$
Which of the following strings is not generated by the grammar?
(a) aaaa (b) baba (c) abba (d) babaaabab.

(iii) A non-deterministic finite state acceptor (ndfsa) M has m> 0 states. M is converted to a deterministic finite state acceptor (dfsa) and then minimized. The resulting machine N has n> 0 states. Then it is always the case that
(a) $m \leq n$ (b) $m = n$ (c) $m \geq n$ (d) $2^m \geq n$.

(iv) How many binary strings of length n, where n> 0 and n is odd, have more 0's than 1's?
(a) 2^{n-1} (b) 2^{n-2} (c) 2^n (d) $2^{(n+1)/2}$.

(v) Let L be an infinite regular set of binary strings. Let L_1 be any infinite proper subset of L. Then the language L_1
(a) necessarily has a Type 3 grammar.
(b) necessarily has a Type 2 grammar but perhaps not a Type 3 grammar.
(c) necessarily has a Type 1 grammar but perhaps not a Type 2 grammar.
(d) might not be recognizable by any Turing machine.

(vi) Consider the regular expression R = (10)*(01)*. Then there is a minimized deterministic finite state acceptor (dfsa) M with *m*> 0 states that accepts all the strings in *R* (and only those strings), where *m* equals
(a) 3 (b) 4 (c) 5 (d) 6.

(vii) The intersection of a context free language (CFL) and a recursive language (R) is
(a) CFL (b) Recursive (c) Regular (d) none of these.

(viii) Consider the following transitions of a Turing Machine (TM)
$\delta(A, a) \rightarrow (B, a, R)$
$\delta(B, b) \rightarrow (C, b, R)$
$\delta(C, a) \rightarrow (C, a, R)$
$\delta(C, b) \rightarrow (D, b, R)$.

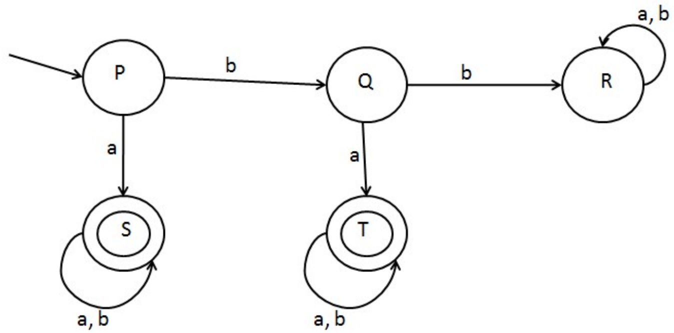
A is the start state and D is the final state. The language accepted by the TM is (in regular expression notation)
(a) aba* (b) aba*ab (c) aba*b (d) a*ba.

(ix) Which automata can be used to check balanced parentheses?
(a) Finite State Automata
(b) Push Down automata
(c) Turing Machine but not Pushdown Automata
(d) (b) and (c) both.

(x) Let L be a context-free language on the alphabet { 0 , 1 }, and let G be a Type 2 grammar for *L* in Chomsky Normal Form. Let α be a binary string of length n> 0 in *L*. Then the derivation tree for α has k internal (i.e., non-leaf) nodes, where *k* equals
(a) n (b) n-1 (c) n+1 (d) 2n.

Group – B

2. (a)



(i) Is this a minimized DFA? If not then minimize it. P is the start state, and S and T are the goal states.
(ii) Find the regular (Type 3) grammar that defines the minimized DFA.

(b) If the final states and non-final states are interchanged for the above DFA, what is the language that is going to be accepted by the DFA?

(4 + 4) + 4 = 12

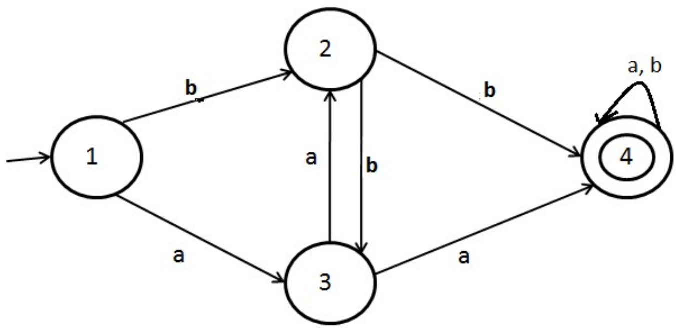
3. (a) Construct a deterministic finite state acceptor (dfsa) M on the input alphabet { 0,1 } that accepts a binary string α if and only if α is contained in the regular expression (01)*(0* +1*).

(b) Construct a deterministic finite state acceptor (dfsa) N on the input alphabet { 0,1 } that accepts a string α if and only if α is not contained in the regular expression (10+01)*.

6 + 6 = 12

Group – C

4. (a)



Find the regular expression corresponding to the automaton given above. 1 is the start state and 4 is the final state.

(b) Consider the language L on the input alphabet ∑= {0, 1} that contains all (and only) strings of the form $0^m1^n0^r$, where m, n, r are three positive integers. Explain why L is a regular language and give a regular expression for L.

7 + 5 = 12

5. (a) State and explain the Pumping Lemma for regular languages.

(b) Use the Pumping Lemma for regular languages to show that the language L = { $0^{2n}10^{3n}$ | n> 0 } is not regular.

4 + 8 = 12

Group – D

6. G (V,T,P,S) is a Type 2 grammar where V = { S,0,1 }, T = { 0,1 }, and P ={ S → 0S (1), S → 0S1 (2), S → 1S0 (3), S → SS (4), S → 0 (5), S → 01 (6), S → 10 (7) }.

(i) Determine L(G), the language of G.

(ii) Show that G is an ambiguous grammar for L(G).

(iii) Modify G and get a new Type 2 grammar G_1 for the same language that is not ambiguous.

4 + 4 + 4 = 12

7. (a) Construct a pushdown acceptor M to accept the context-free language L consisting of all binary strings that contain twice as many 1's as 0's.

(b) Explain clearly how M accepts the string '011101' and rejects the string '00110111'.

8 + 4 = 12

Group – E

8. (a) Two positive integers m and n are written on a Turing machine tape in unary notation. The integer m is to the left of the integer n on the tape, and a single blank cell separates the two integers. At start the read/write head is positioned on the leftmost 1 of m. Give the state transition diagram of a Turing machine M that will compute $m - n$ if $m >= n$, and 0 if $m<n$, and will halt on the leftmost 1 of the result of the subtraction.

(b) Describe how M operates on the following two sets of inputs:
(i) m = 6, n = 3; (ii) m = 2, n = 4.

8 + 4 = 12

9. (a) Define NP problems in terms of Turing machines.

(b) Design a Turing machine M which can accept the language

L= {$a^nb^mc^n$| n ≥ 1 and m ≥ 0}.

Explain how M accepts "aaaccc" but rejects "aabccc".

3 + (6 + 3) = 12

(7) + (5) = 12