Indian Institute of Technology (ISM), Dhanbad
Department of Computer Science and Engineering
Theory of Computation (CSC208)
End Semester, Date: 12 May 2023

Timing: 3 PM - 6 PM | Winter Semester 2022-23 | Max mark: 100

Attempt all questions. Write full justifications for your answers to be evaluated. Write your proofs/constructions in mathematically precise language. Unclear, incomplete, and/or dubious statements would be severely penalized.

1. Consider the following context-free grammar to generate arithmetic expressions in one variable a, involving addition and multiplication operations only. Here, E is the start symbol.

$$E \to E + E \mid E * E \mid a$$

   (a) Draw all the parse trees for the string $a + a * a + a$ following this grammar. **(5)**

   (b) Propose an unambiguous grammar that ensures that the precedence of the '*' over '+' is maintained and operations are evaluated from left to right. Using your grammar draw the parse tree for the expression string in Part (a) of this question. **NOTE: Incorporation of any additional symbol or production in the grammar will induce a heavy penalty.** **(5)**

2. (a) State the pumping lemma for context-free languages. **(4)**

   (b) Use the pumping lemma stated above to show that the language $L = \{a^n b^n c^n \mid n \geq 0\}$ is not context-free. **(6)**

3. Consider the following context-free grammar G in *Chomsky normal form*. **(10)**

$$\begin{aligned} S &\to AB \mid BC \\ A &\to BA \mid a \\ B &\to CC \mid b \\ C &\to AB \mid a \end{aligned}$$

Use the **CYK algorithm** to test the membership of the string '*baaba*' in the language of the given grammar G. Clearly show the working steps. **Note:** Use of any other approach for testing the membership will induce *zero* marks.

4. Consider the following language over $\Sigma = \{a, b, c\}$:

$$L_1 = \{a^i (bc)^j \mid i, j \geq 0 \text{ and } i \geq j\}$$

   (a) Design a Pushdown Automata (PDA) $M = (Q, \Sigma, \Gamma, \delta, s, \perp, F)$ to accept L_1. M must contain *at most two states*. Clearly define the components of the PDA. Draw the transition diagram. Mention clearly whether your machine M accepts by final state or empty stack. **(7)**

   (b) What do you mean by the **instantaneous description of a PDA?** Explain with relevant mathematical notations. **(3)**

P.T.O.

5. Consider the language $L = \{ww \mid w \in \{1\}^+\}$. Design a Turing machine M that generates the corresponding string in L if a w is already on the tape e.g. if $w = 111$ is on the tape then the machine should write 111111 on the tape and halt. Assume that Γ (tape symbols) $= \{1, z\}$ and the tape head is placed on the leftmost '1' on the tape. Your machine should not have more than 4 (four) states.

(a) Clearly write the steps of execution of your Turing machine. (4)

(b) Define the components of your Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, \cancel{b}, F)$, where the notations have standard meaning. (4)

(c) Draw transition diagram for M. (4)

6. Suppose you have a new problem P_2 and you wish to show that $P2$ is *undecidable*. Let P_1 be a known *undecidable problem*.

(a) With the help of a construction, show how you will prove the undecidability of P_2 using P_1. (5)

(b) Clearly state the characteristics of the modules used in your construction. (3)

7. Prove that the class of context-free languages are closed under

(a) Union (4)

(b) Concatenation (3)

(c) Kleene closure (3)

(d) Homomorphism (5)

8. *Consider* a multi-tape Turing machine M with **two** tapes.

(a) State the construction of a single tape multi-track Turing machine M_d, that may be used to simulate the working of M. You are allowed to store composite information in the state of the machine. But that should be finite and well-justified. (5)

(b) Clearly state (with an example if necessary), the working of a single step of the above simulation. (5)

(c) Suppose M runs for t steps on a given input. Use *worst-case complexity analysis* to estimate the run-time complexity of M_d simulating the t steps of M on the same input, in terms of t. (5)

9. Answer the following questions (Unnecessarily long answers will NOT be evaluated.)

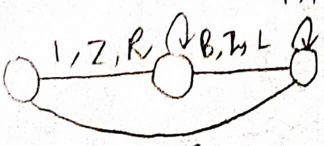
(a) State and discuss the **Church-Turing** thesis. (3)

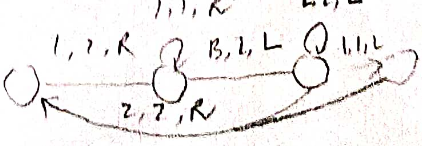
(b) State the difference between *recursively enumerable* and *recursive* languages. (4)

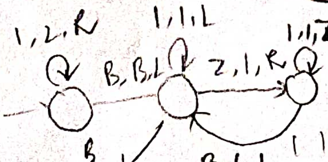
(c) What do you mean by the *non-trivial* property of a recursively enumerable language? (3)

11 | 1|z z

z z







z z z |

z z z|1 1

z z 1 | z z

z z 1 1 | 1 1

z z 1 | 1 1

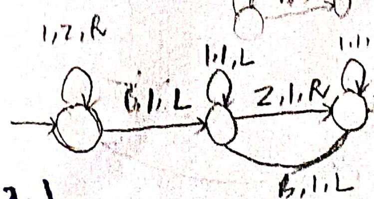
z z z | 1 1 1

b: z z z | 1 1 1

2

1 → z 1

SAT



z z z | 1 1

z z z z | 1 1 1

z z 1 | 1 1