Indian Institute of Engineering Science and Technology, Shibpur

B. Tech. (CST) 4^{4tg} Semester End-Term Examination, May 11, 2022

Theory of Computation (CS 2204)

Full Marks: 50 Time: 90 Min

- **Attempt all questions.**
- **Answers should be precise, to the point, and in your own words as far as practicable.**
- **Make your own assumptions, if necessary, and state them at proper places.**
- **One mark is reserved for clarity of presentation.**

1. Construct an **Automaton** as specified below for **any two** from the following.
   (a) Construct a **Pushdown Automaton** that accepts the language $L = \{\omega \in \{a, b, c\}^* |$ The number of 'a's in ω is same as that of 'b's.$\}$ For example, the strings 'abc', '$cbca$', etc., belong to L.
   (b) Construct a **Turing Machine** of some variation or a **Machine Schema** to compute the function $f : \{a, b, c\}^* \to \mathcal{N}$, where, $f(\omega)$ = the number of 'c's in ω. For example, $f(abcca) = 2$.
   (c) Construct a **Turing Machine** (Deterministic or Non-Deterministic) to **accept** the language $L = \{\omega\omega | \omega \in \{a.b\}^*\}$. For example, the strings '$abab$', '$abbab$b', etc., belong to L.

   [2×8]

2. Construct a **Grammar** (Regular or Context-Free or Unrestricted) as specified below for **any three** from the following.
   (a) Construct a **Regular Grammar** that generates all the valid decimal numbers (integer or fractional) allowed in **C** programming language.
   (b) Construct a **Context-Free Grammar** that generates the language $L = \{\omega \in \{a, b, c\}^* |$ the number of 'a's in ω is half the number of 'b's and 'c's together in $\omega\}$. For example, '$baaccb$ or bba belong to L.
   (c) Construct a **Grammar** that **computes** the function $f : \{a, b\}^* \to \{a, b\}^*$, such that, $f(\omega) = \omega^R$.
   (d) Construct a **Grammar** that **computes** the function $f : \mathcal{N} \to \mathcal{N}$, such that, $f(n) = 2^n$.

   [3×6]

3. Attempt **any three** questions from the following!
   (a) Justify that for any **Nondeterministic Turing Machine** there are only a finite number of computations of some given length.
   (b) Formally justify why every **Context-Free Grammar** qualifies to be an **Unrestricted Grammar**.
   (c) Justify that for every **Pushdown Automaton** M accepting the language L, there exists a **Non-Deterministic Turing Machine** of some variation that accepts L.
   (d) Propose a scheme for representing a **Turing Machine** as a string (after all a Turing machine M is nothing but a 4-tuple (K, Σ, δ, s)). Justify that the set of all such strings (each of which represents a Turing Machine as per your scheme) is a **Context-Free Language**.

   [3×5]