Indian Institute of Technology Kharagpur
Department of Computer Science and Engineering

Test 2

CS41001: Theory of Computation — 5th of October, 2021
Duration = 90 minutes — Total Marks = 50

Answer all questions. State all assumptions you make. Keep your answers concise.
Write on paper, then scan and submit a single pdf file.

1. Write a Type-0 grammar for the language $\{0^{2^i+3^i} \mid i \geq 1\}$. [10]

2. (a) Define *hardness* and *completeness* for the class of recursively enumerable languages with respect to many-one reducibility ($\leq_m$-reducibility) (similar to how you define hardness and completeness for the class **NP** with respect to Karp ($\leq_p$) reduction). [4]

   (b) Provide an example for an *r.e. complete* language. [3]

   (c) Describe a language that is *r.e. hard* but not *r.e. complete*. [3]

3. Given a graph G, two vertices s, t in G and a shortest path P between s and t of length k as input, the $\mathsf{Next-Path}$ problem determines if the shortest path between s and t that is different (at least 1 vertex or edge different) from P has length at most $100k$. Is this problem in **P**? [10]

4. The Nontrivial-SAT problem takes as input a CNF-SAT formula ϕ and determines if there is a satisfying assignment for ϕ such that in each clause at least 1 literal is set to true and at least 1 literal is set to false. Show that Nontrivial-SAT is **NP**-complete.
   Hint: Consider instances where each clause has 4 literals. [10]

5. Assume that SAT cannot have an algorithm running in time $o(2^n)$, where n is the number of variables in the input instance. Prove that under this assumption there are infinitely many languages that are neither in **P** nor **NP**-complete. [10]