6.045J/18.400J: Automata, Computability and Complexity Prof. Ron Rivest

Final Exam - Conflict Version

Please write your name in the upper corner of each page.

Problem 1: **True or False (3 points each)** Full credit will be given for correct answers. You may optionally justify your answer to receive partial credit for an incorrect answer.

1. True or False: Suppose L is a language and M is a probabilistic polynomial-time decider such that
   - $Pr[M(x) \text{ accepts } | x \in L] = \frac{1}{2}$
   - $Pr[M(x) \text{ accepts } | x \notin L] = 0$

   This implies that $L \in BPP$.

2. True or False: If a language is accepted by a 20-state NFA, then it must also be accepted by a 20-state DFA.

3. True or False: The relation $\leq_p$ is reflexive, symmetric and transitive.

4. True or False: Let $L = \bigcup_{i=1}^{\infty} L_i$ and each $L_i \in P$ then $L \in P$.

5. True or False: Consider the following language:

$$\text{FAR} = \left\{ (G, s, t, k) \;\middle|\; \begin{array}{l} G \text{ is an undirected graph with } n \text{ vertices, } s \text{ and } t \text{ are} \\ \text{vertices of } G,\ 0 \leq k \leq n, \text{ and there is no path from} \\ s \text{ to } t \text{ containing } k \text{ or fewer edges.} \end{array} \right\}$$

   $FAR \in NL$.

Problem 2: **(25 points)** Given a DFA M which accepts the language L, show that the language:

$$\text{One-Suffix}(L) = \{x \in \Sigma^* | \text{ There exists exactly one } y \in \Sigma^* \text{ for which } xy \in L\}$$

is also regular.

Problem 3: **(15 points)** Show that the following language is decidable:

$$\text{Star-Subset-Of} = \{< M, N > \mid \text{M and N are DFA's and } L(M)^* \subseteq L(N)^*\}$$

Problem 4: **(25 points)** Let E be an enumerator that enumerates all Turing Machines. Let M_i^E denote the i^{th} Turing Machine output by E. Prove that there exists an i such that

$$L(M_i^E) = L(M_{i+1}^E)$$

That is, show that every such enumerator must output two consecutive Turing Machines which have the same language.

Note: Alternatively, we will give significant credit for showing that there exists an enumerator E that enumerates all Turing Machines such that no three consecutive Turing Machines have the same language.

Problem 5: **(15 points)** Consider the following language:

$$L = \{< M > \mid \text{M is a Turing machine and on some input, w, M prints 6045}\}$$

Show that $A_{TM} \leq_m L$.

Problem 6: **(25 points)** It is well known that if P=NP then there *exists* an efficient algorithm which finds satisfying assignments. However, Leonid Levin claims the stronger claim that if P=NP, then he has found a specific algorithm that on input $\phi \in SAT$, runs in polynomial time and outputs an α such that α is an assignment which satisfies ϕ. (He makes no claims about the algorithm's behavior or run-time when ϕ is not satisfiable).

The algorithm is as follows:

- For t=1 to ∞
  - For k=1 to t
    - For m=1 to t
      - Run the m^{th} Turing Machine for n^k time steps on input ϕ. If the m^{th} Turing Machine outputs β within n^k time steps, then check whether β satisfies ϕ. If β does satisfy ϕ then output β and Halt. Otherwise discard β and continue.
    - Next m
  - Next k
- Next t

Prove or Disprove Levin's Claim.

Problem 7: **(15 points)** Consider the language:

$$EXACT - CLIQUE = \{< G, k > | \text{The largest clique in G is of size exactly k}\}$$

Argue that EXACT-CLIQUE is probably not NP-Complete.

Problem 8: **(15 points)** Select a specific pair of NP-Complete languages, A and B, and show that $A \leq_L B$. (Note: This is not true for every pair of NP-Complete languages, but it is true for most choices of A and B).