College of Computer & Information Science — Spring 2008
Northeastern University — Handout 25
CSG714: Theory of Computation — 24 April 2008

Take-Home Final

Due at 6 PM, Friday, April 25

- **Honor code:** This exam is open-notes, open-library, and open-web. However, *no collaboration of any kind is allowed.* ("Collaboration" includes, for example, discussion or exchange of material related to the problems on the exam with anyone other than the instructor.) If you have any questions or clarifications, please ask me.
- **Policy on Cheating:** Students who violate the above rules on scholastic honesty are subject to academic and disciplinary penalties. Any student caught cheating will receive an **F** (failing grade) for the course and the case may also be forwarded to the Office of Student Conduct & Conflict Resolution for further disciplinary action.
- **Presentation of solutions:** While describing a proof or a solution, you may use any of the proofs and techniques covered in class or in the text by referencing the appropriate material, without elaboration. If you use any other sources, please include citations and write the solutions in your own words.
- **A note on grading:** Your grade on any problem will be determined on the basis of the correctness of your solution and the clarity of the description. Show your work, as partial credit will be given.
- **Best of luck!**

1. (10 points) An alternative definition for deciders

In class, we called a non-deterministic Turing machine a decider if all of its branches halted on all inputs. Define a non-deterministic Turing machine to be a *pseudo-decider* if on every input, either there exists a branch that accepts and halts or all branches halt.

Prove or disprove: the set of languages recognized by deciders is the same as the set of languages recognized by pseudo-deciders.

2. (7 + 6 + 7 = 20 points) Checking for subsequences

Consider the following language A:

$$\{x\#y : x, y \in \{0,1\}^* \text{ and } x^R \text{ is a subsequence of } y\},$$

where x^R denotes the reverse of string x. We say that a string $u_1 u_2 \ldots u_n$ is a *subsequence* of a string $v_1 v_2 \ldots v_m$, where $u_i, v_j \in \{0,1\}$, $1 \le i \le n$, $1 \le j \le m$, if there exist n integers j_1 through j_n such that $1 \le j_1 < j_2 < \ldots < j_n \le m$ and $u_i = v_{j_i}$ for $1 \le i \le n$.

For each of the following parts, answer "yes", "no", or "don't know". Justify your answer in each case.

(a) Is A regular?

(b) Is A context-free?

(c) Is A in NL?

3. (10 points) Undecidability of RP machines

We have defined a language A to be in RP if there exists a probabilistic Turing machine M such that: if w is in A then M accepts w with probability at least 1/2; otherwise, M always rejects w.

An alternative way to define RP is as follows. A language A is in RP if there exists a nondeterministic polynomial-time Turing machine M such that on any input w: if w is in A then at least half of the computations of M on w accept; otherwise, all computations reject.

Show that it is undecidable to determine whether a given nondeterministic Turing machine M satisfies the following condition: it runs in polynomial time and on every input either all computations reject or at least half of them accept.

4. (10 points) Constructing a satisfiable assignment

Suppose we are given a Boolean formula ϕ with *less than* n^k clauses, each clause having $\lceil k \log_2 n \rceil$ literals such that no two literals in a clause correspond to the same variable. (Assume k is a fixed integer.) Using the probabilistic method (similar to one we used for MAX3SAT), show that there exists a satisfiable assignment for ϕ. Then, present a deterministic algorithm that runs in time polynomial in n and returns a satisfiable assignment for ϕ.

5. (10 points) Finding paths in a graph

We are given an undirected graph G and k mice at k starting nodes s_1 through s_k, respectively, and wanting to go to end-nodes e_1 through e_k, respectively. Show that it is NP-complete to decide whether there exists a path for each mouse such that no two paths share a node.

Hint: One possibility is to use 3SAT for your reduction. For a formula with n variables and m clauses, set $k = n + m$. For each variable v and each clause c in the formula, construct two vertices s_v and t_v, (respectively s_c and t_c). Depending on whether the variable v (or its negation) is in the clause c, add nodes x_{vc} (respectively y_{vc}).

Bonus Problem (4 points)

Give the weakest condition you can give on $r(n)$ and $q(n)$ to make PCP$(r(n), q(n))$ equal to P.