Princeton University — Fall 2002
COS 487: Theory of Computation — Prof. Amit Sahai

Final Examination

Released: December 26, 2002 *Due: January 15, 2003* *TA: Manoj M P*

- This exam has five questions. You have **48 hours** to take this exam. You can choose any contiguous 48 hours and work on the exam during that time. Do not read the problems before that.
- The exam is **open-book**, which means that while taking the exam you can refer to the text-book (Sipser), the class-notes and your graded assignments. But you shall not refer to any other material, in particular, other books or internet resources.
- **No collaboration** with anyone is allowed. You shall work independently on all the problems.
- Print out this file and write your solutions in the space provided. Limit your solutions to **one page** (the space below the problem).
- In case you need clarification on any problem you can send an e-mail to the TA (allow at least 24 hours for reply).

Please write your name and year/class below:

Name: ________________________ Year: ______

Please write the start and finish time below. Write the start-time before you start. Do not work on the exam after you note down the finish-time.

Started: Date ____________ Time ____________

Finished: Date ____________ Time ____________

- Write out and sign the Honor Code pledge before turning in the test.

"I pledge my honor that I have not violated the Honor Code during this examination."

Best of Luck!

1. A graph is k-colorable if there is a way to color the vertices of the graph with k colors such that no edge has both vertices colored with the same color. You might know that deciding if a graph is 3-colorable or not is NP-complete. In this problem you shall show that 2-colorability is a much simpler problem.

   The language 2COLOR is defined as $\{G|G \text{ is a 2-colorable graph}\}$. Show that 2COLOR $\in \mathsf{NL}$.

   [*Hint*: A graph is called bi-partite if the vertex set V can be partitioned into two parts as $V = A \cup B$, such that for every edge $\{u, v\}$, $u \in A$ iff $v \in B$. You can use the fact (without proof) that a graph is bi-partite iff it has no odd length cycles.]

   (*Partial credit for proving that* 2COLOR $\in \mathsf{P}$.)

2. Let $\mathsf{A}_{\mathsf{NFA}} = \{\langle M, w\rangle \,|\, M$ is an NFA and M accepts $w\}$. Show that $\mathsf{A}_{\mathsf{NFA}}$ is NL-complete.
   [*Hint*: This problem is easy.]

3. (This is Problem 9.10 from Sipser)

Show the error in the following fallacious "proof" that $\mathsf{P} \neq \mathsf{NP}$. Proof by contradiction. Assume that $\mathsf{P} = \mathsf{NP}$. Then $SAT \in \mathsf{P}$. So for some k, $SAT \in \mathsf{TIME}(n^k)$. Because every language in NP is polynomial time reducible to SAT, $\mathsf{NP} \subseteq \mathsf{TIME}(n^k)$. Therefore $\mathsf{P} \subseteq \mathsf{TIME}(n^k)$. But by the time hierarchy theorem, $\mathsf{TIME}(n^{k+1})$ contains a language which is not in $\mathsf{TIME}(n^k)$, which contradicts $\mathsf{P} \subseteq \mathsf{TIME}(n^k)$. Therefore $\mathsf{P} \neq \mathsf{NP}$.

4. Suppose we modify the definition of push-down automata (PDA) by replacing the stack by a queue. (A queue is a "First In- First Out" structure, as opposed to a stack which is a "Last In- First Out" structure). Show that the modified PDAs can simulate Turing Machines.

5. Recall that we say a language L is *reducible* (or more formally, polynomial time Turing reducible) to a language L' if there exists a *polynomial time* machine $M^{L'}$ which makes oracle queries to L' (i.e., it can make a query y and will receive as answer whether $y \in L'$ or not) and decides L.

   A language L is *downward self-reducible* if there exists a polynomial time machine M^L, such that M^L (M given access to an oracle deciding L) can decide L, but under the condition that on input x, it makes queries only of length *strictly less than* $|x|$.

   Show that every downward self reducible language is in PSPACE.