COS 487: Theory of Computation **Fall 2008**

Assignment #4

Due: Tuesday, November 18 *Sanjeev Arora*

1. Show that $\mathcal{P}$ is closed under the star (Kleene closure) operation.

(*Hint:* Use dynamic programming. Say $L \in \mathcal{P}$. Given a string $y_1 \dots y_n$, maintain, for every pair of indices $i \leq j$, a boolean $A[i,j]$ that indicates if $y_i \dots y_j$ is in L^*).

Describe precisely why your algorithm takes polynomial time.

2. Show that $\mathcal{NP}$ is closed under the star operation.

3. Show that the following language is NP-complete.

$$\{\langle M, x, 1^t \rangle \ : \ \text{NDTM } M \text{ accepts } x \text{ in } t \text{ steps}\}$$

4. A **cut** in an undirected graph is a separation of the vertices into two disjoint subsets S and T. The *size* of the cut is the number of edges with one end-point in S and the other in T. Let

$$MAXCUT = \{\langle G, k \rangle \ : \ G \text{ has a cut of size } k\}$$

Show that $MAXCUT$ is NP-complete.

You may use the reduction from $\neq SAT$, defined in problem 7.23 of the book. Refer to the book also for a hint on the reduction (Problems 7.23-24 of the new edition and 7.22-23 of the first edition).

5. This problem is inspired by the single-player game Minesweeper, generalized to an arbitrary graph. Let G be an undirected graph, where each node either contains a single, hidden mine or is empty. The player chooses nodes, one by one. If the player chooses a node containing a mine, the player loses. If the player chooses an empty node, the player learns the number of neighboring nodes containing mines. (A neighboring node is one connected to the chosen node by an edge). The player wins if and when all empty nodes have been so chosen.

In the mine consistency problem you are given a graph G, along with numbers labeling some of G's nodes. You must determine whether a placement of mines on the remaining nodes is possible, so that any node v that is labeled m has exactly m neighboring nodes containing mines. Formulate this problem as a language and show that it is NP-complete.

6. Suppose there exists a polynomial time algorithm that can decide if a boolean formula is satisfiable. Show how to use this algorithm to find a satisfying assignment to a given boolean formula. Explain clearly why the running time is polynomial.

Optional. Do the same for the graph isomorphism problem. Suppose you have a polynomial time algorithm that can decide if two graphs are isomorphic. Use this to design a polynomial time algorithm that actually finds the isomorphism.

7. Let $f : \mathbb{N} \to \mathbb{N}$ be a function that is $o(\log \log n)$. Let L be a language in $SPACE(f(n))$. Show that L is in fact regular. (Note that regular languages are in $SPACE(O(1))$).

Hint. A good way to think about space complexity is to imagine a TM with a read-only tape consisting of the input and an additional "work tape". The size of the work tape counts for the space complexity. Now, consider the shortest string s in L that *requires* space of m bits. Consider the 'configurations' the machine could have been in as it 'processed' the different substrings of s. Use minimality of s to claim they should all be different, hence the length of s is at most ...

Optional. Consider the language $\{[1]_2\#[2]_2\#[3]_2\#\ldots\#[n]_2 \ : \ n \geq 1 \text{ is an integer}\}$, where $[n]_2$ denotes the binary representation of n, and the language is over the alphabet $\{0, 1, \#\}$. Show that this language is in $SPACE(O(\log \log n))$, but is not regular.

8. Say that two Boolean formulas are *equivalent* if they have the same set of variables and are true on the same set of assignments to those variables (i.e., they describe the same Boolean function). A Boolean formula is *minimal* if no shorter Boolean formula is equivalent to it. Let *MIN-FORMULA* be the collection of minimal Boolean formulas.

(a) Show that *MIN-FORMULA* is in PSPACE.

(b) Explain why the following does not show *MIN-FORMULA* $\in$ co-$\mathcal{NP}$: If $\phi \notin$ *MIN-FORMULA*, then ϕ has a smaller, equivalent formula. An NTM can verify that $\phi \in \overline{\textit{MIN-FORMULA}}$ by guessing that formula.

9. Let B be the language of properly nested parentheses and brackets. For example, `([()()]()[])` is in B but `([)]` is not. Show that B is in LOG-SPACE.