CS 172: Computability and Complexity, Spring 2008	S. A. Seshia & A. Kulkarni

Midterm 2

April 17, 2008

YOUR NAME:

Instructions:

This exam is *closed-book, open-notes*. Please turn off and put away electronic devices such as cell phones, laptops, etc.

You have a total of 75 minutes. There are 4 questions worth a total of 100 points. The questions vary in difficulty, so if you get stuck on any question, it might help to leave it for a while and try another one.

Answer each question in the space provided below the question. If you need more space, you can use the reverse side of that page. *You can use without proof any result proved in class, in Sipser's book, or on homeworks, but clearly state the result you are using.*

Do not turn this page until the instructor tells you to do so!

Problem 1	
Problem 2	
Problem 3	
Problem 4	
Total	

Problem 1: [True or False, with justification] (30 points)

For each of the following four questions, state TRUE or FALSE. Justify your answer with a short proof or simple counterexample.

(a) Recall the proof of the Cook-Levin theorem, where we used *legal* 2×3 windows. Suppose that the transition function δ of the TM N we are encoding is such that $\delta(q_1, a) = (q_2, b, L)$.

Then, the following 2×3 window is legal for machine N.

b	a	b
b	b	b

(b) For any $n > 0$, there exists a set of languages $L_1, ..., L_n$ such that, for all i, $L_i \in NP$ and $L_1 \leq_P L_2 \leq_P ... \leq_P L_{n-1} \leq_P L_n \leq_P L_1$.

(c) Unless P=NP, every language decidable by a non-deterministic Turing Machine in polynomial time requires at least exponential time on a deterministic TM.

Problem 2: (20 points)

Let $B = \{\langle M_1, M_2 \rangle \mid L(M_1) = \overline{L(M_2)}\}$. Prove that B is not Turing-recognizable.

[Hint: Try a reduction from $\overline{A_{TM}}$.]

Problem 3: (20 points)

A *useless state* in a PDA is a state that is never entered on any input string. Let $L = \{\langle P \rangle \mid P \text{ is a PDA that has useless states }\}$. Prove that L is decidable.

[Hint: Use a reduction to E_{CFG}, which we proved to be decidable.]

Problem 4: (30 points)

For a graph G, define a *path* from v_1 to v_m to be a sequence of vertices $v_1, v_2, \ldots, v_m$ where (v_i, v_{i+1}) is a edge and each vertex appears at most once.

Recall from class that a Hamiltonian Path on a graph G is one which visits *all vertices* of G *exactly once*. The language $HAMPATH$ can be given as:
$HAMPATH = \{\langle G, s, t\rangle | G$ is an directed graph with vertices s and t, with a Hamiltonian path from s to $t\}$.

Sipser's book includes a proof that $HAMPATH$ is NP-complete. You can use this result.

In this question, we investigate two related problems. (Be sure to turn the page for the second part!)

(a) Consider the problem of finding the longest path between two vertices in a directed acyclic graph:
DAGPATH $= \{\langle G, k, s, t\rangle \mid G$ is a directed acyclic graph with a path of length at least k vertices from s to $t\}$

Show that DAGPATH $\in$ P.

(b) Now consider the problem of finding the longest path between two vertices in an arbitrary directed graph:
LongestPATH $= \{\langle G, k, s, t\rangle \mid G$ is a directed graph with a path of at least k vertices from s to $t\}$

Show that LongestPATH is NP-complete.