CS172 Computability and Complexity **Spring 2011**

Final Exam

May 10, 2011

Your name and SID:

Instructions:

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

You have a total of 170 minutes. There are 6 questions worth a total of 150 points plus a bonus question worth of 15 points. However, your final grade will not exceed 150. The questions vary in difficulty, so if you get stuck on any question, it might help to leave it 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 during discussion sections, but clearly state the result you are using. Descriptions of Turing machines can be in the form of Sipser's "high-level descriptions." Show your work in all proofs.

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

Problem 1 (50 pts)		Problem 5 (25 pts)	
Problem 2 (10 pts)		Problem 6 (20 pts)	
Problem 3 (30 pts)		Problem 7 (15 pts)	
Problem 4 (15 pts)		Total (150 pts)	

Problem 1: [True of False, with justification] (50 points)
For each of the following questions, state TRUE or FALSE, and justify your answer.

a) The language $\{\langle D_1, D_2\rangle \mid D_1 \text{ and } D_2 \text{ are DFAs and } L(D_1) = L(D_2)\}$ is in P.

b) If NL $=$ L then L $=$ NL-complete.

c) There exists *no* language A such that, for all languages B, we have $B \leq_M A$.

d) Let $A_1 \leq_L A_2$ be two languages. If $A_2 \in \mathrm{TIME}(t(n))$ for some function $t(n)$ then A_1 can be decided by a Turing machine in time $O(t(n))$.

e) The language $\{\langle D\rangle \mid D \text{ is a DFA and } L(D) = \emptyset\}$ is NL-complete.

Problem 2: (10 points)
Show how SUBSET-SUM can be reduced in polynomial time to 0-1 integer programming. Recall that given an $m \times n$ matrix A and a vector b with n elements, 0-1 integer programming is the problem of determining whether there exists a binary vector x with n elements such that $Ax \leq b$, where the inequality is satisfied element-wise.

Problem 3: (30 points)
Consider the following language over some fixed alphabet Σ:

$$A = \{\langle M_1, M_2, k\rangle \mid M_1 \text{ and } M_2 \text{ are TMs and } L(M_1) \cap L(M_2) \text{ has at least } k \geq 1 \text{ strings}\}.$$

a) Show that A is Turing-recognizable.

b) Show that A is not Turing-decidable.

Problem 4: (15 points)
Give the regular expression for the set of strings over the alphabet $\{0, 1\}$ which do *not* contain 101 as a substring.

Problem 5: (25 points)

a) Let D_1 and D_2 be DFAs over the same alphabet. Let m_1 be the number of states of D_1 and m_2 be the number of states of D_2.
Prove that, if $L(D_1) \neq L(D_2)$, then there exists $x \in \Sigma^\star$ such that $|x| \leq m_1 m_2$ and $x \in (L(D_1) \cup L(D_2)) \setminus (L(D_1) \cap L(D_2))$.

b) Show that

$$\{\langle N_1, N_2\rangle \mid N_1 \text{ and } N_2 \text{ are NFAs over the alphabet } \{0,1\} \text{ and } L(N_1) \neq L(N_2)\}.$$

is in PSPACE.

Problem 6: (20 points)
Consider the language

$$A = \{\langle G, w\rangle \mid G \text{ is a linear grammar, } w \text{ is a string and } w \in L(G)\}.$$

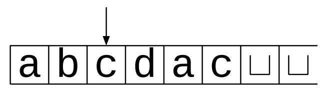
Show that $A \in$ NL-complete.
Recall that a grammar is called linear if the right hand side of each rule has at most one variable; that is, all the rules of a linear grammar have the form $X \to \alpha Y \beta$ where X and Y are variables and $\alpha, \beta \in \Sigma^\star$ are strings of terminals.

Problem 7: (Bonus question, 15 points)

We define a *FIFO automaton* as a PDA where the stack is replaced by a FIFO (first-in-first-out) queue. That is, at each transition, a FIFO automaton can pop the *first* element of the queue and push an element to the *end* of the queue.

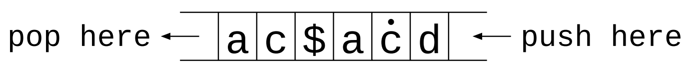
Show that a Turing machine can be simulated using a FIFO automaton.

Hint: If the Turing machine is in state q with tape content



TM tape

then the FIFO automaton is in state (q, b) with queue content



FIFO queue

(Note that the symbol b is *not* in the queue.)

(extra page)

(extra page)