3261 Final
May 7, 2007

Prof. Tal Malkin

Closed book exam, two pages of notes allowed.
Write all answers in the space provided. If you run out space use the back of the page. If you need scratch paper, ask a TA or the professor.
Exam starts at 4:10pm and ends at 7:00pm.
Total no. of points is 126.
Write your name and email address on the top of any *loose* page.
Explain all your work, and state any assumption you use in your solutions. If you use theorems proved in class, you should explicitly state what you are using.
It is highly recommended that you read the entire exam before proceeding as questions are ordered by topic, not by difficulty.

Name: ______________________________

Email: ______________________________

1a(6)	1b(6)	1c(6)	1d(6)	2a(4)	2b(12)	2c(12)	3a(8)	3b(12)	3c(6)	4a(12)	4b(6)	4c(12)	5(18)	Total(126)

1. **Short Answers**

   For each of the following, indicate whether the claim is true or false. If true, prove it. If false, provide a counter example and justify it.

   (a) (6 points) If L is finite, then $\overline{L}$ (namely the complement of L) is regular.

   (b) (6 points) If L_1 is not recognizable, and $L_1 \leq_m L_2$, then L_2 is not decidable.

(c) (6 points) For any two languages L_1, L_2, either $L_1 \leq_m L_2$ or $L_2 \leq_m L_1$.

(d) (6 points) If L is in P and $L' \subseteq L$, then L' is also in P.

2. **The Language of Regular Expressions**

   Recall that every regular expression is a string (which happens to represent a regular language). Thus, we may consider the language $\mathbf{L_{REX}}$ consisting of all regular expressions over $\{0,1\}^*$. For example, the strings $(((0 \cup 1)^* \cup (\emptyset)^*) \circ 0)$, $(00)^*$, and $112 \cup 01^*$ are in L_{REX}. In this problem you will prove that L_{REX} is context free, but not regular.

   (a) (4 points) What is the alphabet over which the language L_{REX} is defined?

   (b) (12 points) Prove that L_{REX} is not regular.

   Hint: it may be helpful to denote by A the string (0 and by B the string 0), and to consider the language A^*B^*.

(c) (12 points) Show a context-free grammar for L_{REX} (i.e., a CFG that generates all regular expressions, and nothing else).

3. **Variants of Turing Machines**

Consider the following variant of a Turing-Machine, which we call *Probabilistic Turing Machine with linear randomness* (PTM). A PTM is defined in a similar way to a standard TM, except that it receives input on two tapes: The first tape is referred to as the *input-tape*, and the second is referred to as the *random-tape*.

The input-tape is used exactly as in a standard TM. Namely, at the beginning of the computation it contains the input x for the machine (and there is a read-write head which points to the first symbol of x). The random-tape is a read-only tape, which at the beginning of the computation contains a string r of 0's and 1's of the same length as x. For this tape there is a read-only head which at the beginning of the computation points to the first symbol of r.

We say that a string x (of length n) is in the language of a PTM M, if for more than half of the possible strings r of length n, M accepts the pair (x, r). That is, from the start state with x on the input-tape and r on the random tape, M eventually enters the accepting state.

In other words, the language recognized by a PTM M is formally defined as

$$L(M) = \{x \mid \text{there are more than } 2^{|x|-1} \text{ strings } r \text{ of length } |x| \text{ for which } M \text{ accepts } (x, r)\}$$

where $|x|$ is the length of the string x.

Remark: Notice that for an input x, the machine M can have many different computation paths, depending on the value of r. We can think of r as being a "guess" of the machine as to which path to take. This is somewhat similar to the "guesses" that a non-deterministic TM can make. The difference is that for x to be in the language of M, we need that more than half of the guesses will lead to acceptance (whereas in a non-deterministic machine we only need one accepting guess).

(a) (8 points) Prove that if a language is recognized by a standard TM, then it is also recognized by some PTM (A high-level description is sufficient).

(b) (12 points) Prove that if a language is recognized by a PTM, then it is also recognized by some standard TM. (A high-level description is sufficient).

(c) (6 points) Give a rough estimate for the time it takes a standard TM to simulate a PTM. That is, if for every r the PTM halts on (x, r) after at most T steps, then (approximately) how many steps does the standard TM run on x?

4. **Decidability and Recognizability**

   Recall that we defined $\mathrm{X}_{TM} = \{\langle M \rangle : M \text{ does not accept } \langle M \rangle\}$ and proved that it is not decidable, and not even recognizable. Define the language

$$\mathrm{NE}_{TM} = \{\langle M \rangle : L(M) \text{ is not empty}\}.$$

   (a) (12 points) Prove that NE_{TM} is undecidable, by showing that $\mathrm{X}_{TM} \leq_T \mathrm{NE}_{TM}$. Specifically, let N be a decider for NE_{TM}, show a decider T for X_{TM}.

   (b) (6 points) Now assume N is a recognizer (rather than a decider) for NE_{TM}. Is the TM T you constructed in part (a) a recognizer for X_{TM}? Explain your answer.

(c) (12 points) Prove that NE_{TM} is recognizable.

5. **(18 points) NP-Completeness**

   (18 points) Prove that the language

   DOUBLE-SAT=$\{\langle\phi\rangle : \phi$ is a Boolean formula with at least two satisfying assignments$\}$

   is NP-complete.