**6.045J/18.400J: Automata, Computability and Complexity** Prof. Ron Rivest

# Handout 10a: Alternate Quiz 2

Write your full name on each page.

Name: ______________________________

**Problem 1: Short Answer: (3 points each)**

1. True or false: The intersection of two decidable languages is decidable. Briefly state a reason for your answer.

2. Suppose a *randomized* TM $M$ had the transition function

$$\delta : Q \times \Gamma \times \{0, 1\} \to Q \times \Gamma \times \{L, R\}$$

That is, it could flip a coin at every transition, and the transition depended on the current state, the contents of the current tape cell, and the coin flip. Now, instead of a machine simply accepting or not accepting a word $w$, it would have a probability of accepting. Is the language

$$\{w \ : \ \text{Probability that } M \text{ accepts is greater than } 0\}$$

recognizable, decidable, or neither? Briefly state a reason for your answer.

3. We say that a Turing machine $M$ has *maximal* length if no other Turing machine equivalent to $M$ has a larger description. Is the following language decidable, recognizable, or neither? Explain your answer.

$$MAX_{TM} = \{\langle M \rangle \ : \ M \text{ is maximal}\}$$

4. True or False: Every countably infinite language is recognizable. Briefly explain your answer.

5. True or false: If $A \leq_m B$ and $B$ is regular, then $A$ is regular. Briefly state a reason for your answer.

Name:

**Problem 2**: **(15 points)** Prove that there exists a Turing Machine $M$ that accepts only Turing machines with fewer states than $M$ itself. (Hint: recursion theorem.)

Name:

**Problem 3**: **(15 points)** Let

$$L = \{\langle M_1, M_2 \rangle \ : \ M_1 \text{ and } M_2 \text{ are Turing machines and } L(M_1) \cap L(M_2) = \emptyset\}.$$

Show that $L$ is undecidable.

**Problem 4**: **(15 points)** Show every infinite recognizable language has an infinite decidable sub-language. (A language $L$ is a sublanguage of a language $L'$ if $L \subseteq L'$.) Hint: recall that a language is decidable iff there exists an enumerator that enumerates it in lexicographic order.

Name:

**Problem 5**: **(15 points)** A *queue* machine is like a Turing machine, except that it has a queue instead of a tape. It has a finite queue alphabet $\Gamma$ and a finite input alphabet $\Sigma \subseteq \Gamma$. If $x \in \Sigma^*$ is the input, then the machine starts in its start state with $x\$$ in the queue (where \$ is a special symbol in $\Gamma \setminus \Sigma$. The \$ symbol is at the *end* of the queue.) In each step, it removes a symbol from the front of the front of the queue. Based on that symbol and the current state, it pushes a string $z \in \Gamma^*$ onto the back of the queue and enters a new state according to its transition function. It accepts by emptying the queue.

Show that a queue machine is as powerful as a Turing Machine.