CS 364 Exam 1

Please complete this section:

Name (please print): ______________________________

Student Number: ______________________________

No aids allowed. There are 4 questions worth a total of 50 marks. Answer all questions on the question paper. Use backs of pages for scratch work. Check that your exam has 6 pages.

1. (10 points, 5 each)

   (a.) Define the class of recursive languages over the alphabet $\{0, 1\}$. You may assume the definition of a Turing machine, and the definitions of what it means for a Turing machine to (i) halt and accept, (ii) halt and reject, (iii) not halt on a given input.

   A language L over $\{0, 1\}$ is recursive if there exists a Turing machine M such that: (i) M halts on all inputs, and (ii) for all $x \in L$, M halts and accepts on input x, and (iii) for all x not in L, M halts and rejects on input x.

   (b.) Define the class of recursively enumerable languages over $\{0, 1\}$.

   A language L over $\{0, 1\}$ is recursively enumerable if there exists a Turing machine M such that: (i) for all $x \in L$, M halts and accepts x, and (ii) for all x not in L, M either does not halt on x, or M halts and rejects on input x.

2. (10 points) Give a proof sketch showing that any finite language over the alphabet $\{0, 1\}$ is recursive. As a first step, you may want to consider a particular finite language, such as $L = \{1, 11, 111\}$ and show that this language is recursive. Then try to generalize to an arbitrary finite language. (I am looking for a high-level algorithm only.)

Let L be some finite language, and assume there are exactly k strings in L. Then we can create a TM that just examines the input string and accepts it if and only if it is one of the k strings in L. Since the k strings are finite, and each string is finite, the number of states will be finite as well. If we tried to do the same thing to recognize an infinite language, it would require an infinite number of states.

3. (20 points, 10 each) For each of the following languages, indicate the status of this language by specifically indicating whether

   - The problem is recursive.
   - The problem is recursively enumerable but not recursive.
   - The problem is not recursively enumerable

   Justify your answer. If the language is recursive, briefly describe an algorithm that always halts and accepts the language; if the language is recursively enumerable but not recursive, describe a TM/algorithm that accepts the language, and use a reduction to show that it is not recursive. Finally, if the problem is not recursively enumerable, give a reduction to show this. For your reductions, you may use any problem from the standard list of languages given at the end of this examination.

   (a.) The language L consists of encodings $< M, w >$ such that M on input w halts within $|w|$ steps.

   This language is recursive. We simply keep a counter, and simulate M on w for $|w|$ many time steps. Then the pair $< M, w >$ is accepted if during this simulation, M halts.

(b.) The language L consists of encodings $< M >$ of Turing machines such that M accepts *at least one* string w, where w is a binary string beginning with '111'. (M can accept other strings as well.)

The above language is re but not recursive. To see that it is re, we use dovetailing. That is, we enumerate all strings beginning with '111. Then we simulate M on the first i strings for i steps, for increasing i. Halt and accept if M ever accepts any of the strings during any of the simulations. If there is some string beginning with '111' that is accepted by M, then eventually the dovetailing procedure will halt and accept $< M >$.

To see that L is not recursive, we will reduce HB to L as follows. If x is not a valid TM encoding, then $f(x) = x$. Otherwise, assume that $x =< M >$. Create a new machine M' where M' ignores its input, y, and simulates M on the blank input. M' accepts y if and only if M halts on the blank input during the simulation.

The function f is computable, and x is in HB if and only if $f(x)$ is in L.

4. (10 points) State whether each of the following statements is True of False. You will receive two points for each correct answer, 0 points for each incorrect answer, and a half of a point for each question that you leave blank.

(a.) For all languages L_1 and L_2, if both L_1 and L_2 are recursive, then $L_1 \cup L_2$ is recursive.

True

(b.) For all languages L, if L is recursive, then $\overline{L}$ (the complement of L) is recursively enumerable.

True

(c.) For all languages L, if L is recursive, then $\overline{L}$ is recursive.

True

(d.) The set of all infinite strings over $\{0, 1\}$ is countable.

False

(e.) The set of all languages over the alphabet $\{0, 1\}$ is uncountable.

True

LIST OF PROBLEMS THAT ARE NOT RECURSIVE

A_{TM}
Instance: $< M, w >$, the encoding of a Turing machine followed by an input w.
Question: Does M accept input w?

HB
Instance: $< M >$, the encoding of a Turing machine.
Question: Does M halt on the all blank tape?

LIST OF PROBLEMS THAT ARE NOT RE

INF
Instance: $< M >$, the encoding of a Turing machine.
Question: Does M accept an infinite language?

L_d
Instance: w, a binary string.
Question: If w is the i^{th} string (in lexicographic ordering) does the i^{th} Turing Machine (in lexicographic ordering), M_i, NOT accept w?