UR Honor Pledge for Exams: “I affirm that I will not give or receive any unauthorized help on this exam, and that all work will be my own.” (UR now requires you to by hand copy the above sentence onto the blank lines below and then sign it.)

Instructions: There are six problems here. Do four. If you choose to answer more than four problems (you may if you want to), we will grade all that you answer, and will count the highest (that is not a typo; it really says and means “highest”) four scores.

The highest possible score on this exam is 100%, i.e., 100/100 (see, at your leisure after the exam, the 160926 “Midterm I Tidbit,” which is related to this.)

Just as on quizzes, no questions are allowed during this exam.

Throughout this exam, HP denotes \{ i \mid M_i(i) \text{ accepts} \}. Note carefully that the word there is not “halts” but is “accepts.”

Don’t just jot down random stuff (doing so may lose points)—think before answering (we’ll hand out scrap paper on which to doodle/plan/try things).

Note that not all questions are necessarily identical in difficulty, so don’t, for example, make the mistake of spending all your time on one hard problem and leaving the others blank.

Don’t confuse “from” and “to” on reductions (if when trying to prove \( A \leq_m B \) you accidentally just prove \( B \leq_m A \), you’ll probably get no points). Note, for example, that there is a many-one reduction from \( \emptyset \) to HP, but there exists no many-one reduction from HP to \( \emptyset \). (On this exam, “many-one” of course refers to recursive many-one reductions.)

Your handwriting must be clear and readable. We will not guess that some unclear character is what is needed to make your answer correct; your written characters must be clearly readable on their own.

Rules: Closed book, closed notes, no computers or calculators, we prefer pen but you may if you wish use pencil, however, the only colors you may use of pen or pencil are blank and blue. An exception to the “closed notes” rule is that each person can have one 2-sided sheet (or two 1-sided sheets) of self-prepared, hand-written notes.

Make sure to put all your answers that you want graded onto (not the scrap paper but rather) the stapled test sheets.

Scoring: To repeat: There are six problems here. Do four. If you choose to answer more than four problems (you may if you want to), we will grade all that you answer, and will count the highest four scores. Since your grade will be based on four 25-point problems, there are 100 points available here.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Total (of highest 4)</td>
<td></td>
</tr>
</tbody>
</table>
Question 1  [25 points] Prove by a clear Tarski-Kuratowski quantifier analysis that 
\[ \{ i \mid L(M_i) \neq \Sigma^* \} \] is in \( \Sigma_2 \).
Answer: This set can be described as follows:
\[ \{ i \mid (\exists x)(\forall t)[M_i \text{ does not accept } x \text{ within } t \text{ steps}] \} \].
The longest series of alternating quantifiers is \( \exists \forall \), so the set is in \( \Sigma_2 \).
Question 2 [25 points] Let \( A = \{ i \mid L(M_i) = \emptyset \} \). Let \( B = \{ (j, k) \mid L(M_j) = L(M_k) \} \). Give an explicit, direct, many-one reduction from \( A \) to \( B \) (i.e., \( A \leq_m B \)), and prove (in our 3-part way) that it is indeed is such a reduction.

Answer:

The function \( \sigma \) will provide such a reduction. Fix an integer \( j_0 \) such that \( L(M_{j_0}) = \emptyset \). Let \( \sigma(i) = (i, j_0) \). \( \sigma \) is clearly a recursive function.

If \( i \in A \), then \( L(M_i) = \emptyset \), and so \( L(M_i) = L(M_{j_0}) \). Thus \( \sigma(i) = (i, j_0) \in B \).

If \( i \notin A \), \( L(M_i) \neq \emptyset \). Thus \( L(M_i) \neq L(M_{j_0}) \) and \( \sigma(i) = (i, j_0) \notin B \).

We have that for each \( i \) it holds that \( i \in A \Leftrightarrow \sigma(i) \in B \), and \( \sigma \) is a recursive function, so we have that \( A \leq_m B \).
Question 3 [25 points] Let $R = \{ i \mid L(M_i) \neq \emptyset \}$. First, state Rice’s Theorem (the easy 1-part theorem, not the 3-part theorem). Second, either use that Rice’s Theorem to prove that $R$ is not recursive, or explain clearly why that Rice’s Theorem is not appropriate for this problem.

Answer:

Rice’s Theorem states that any nontrivial property of the RE languages is undecidable.

Rice’s Theorem applies to this set. This is a set of TMs defined based on a property of the languages of the TMs, and it is a nontrivial property, as there are RE sets that possess the property and there are RE sets that do not. The set $\Sigma^*$ possesses this property, and the set $\emptyset$ does not. Thus by Rice’s Theorem (the 1-part one) this set is not recursive.
Question 4 [25 points]
(a) [12.5 points] Prove that $\Sigma_1 \cap \Pi_1 = \Sigma_0$ (i.e., prove that if $A$ is both r.e. and co-r.e., then $A$ is recursive).

Answer:
Let $A$ be a set that is both RE and coRE. Thus there exist a TM accepting $A$ and a TM accepting $\overline{A}$. We can use these to construct a TM for $A$ that will always halt (i.e., a TM that \textit{decides} $A$—not merely accepts $A$, but outright decides $A$), thus showing that $A$ is recursive.

Our TM will work as follows. On input $x$, run in a dovetailed fashion the TMs for $A$ and $\overline{A}$ (i.e., run one for one step, then the other for two steps, then the first for three steps, then the second for four steps, and so on and on). If we see the TM for $A$ accept, then halt and accept. If we see TM for $\overline{A}$ accept, then halt and reject. On each string, exactly one of the TMs would eventually accept (because the string has to be either in or not in $A$, and that “or” is in fact an exclusive or). We will thus halt and accept on every member of $A$, and we will halt and reject on every member of $\overline{A}$. We then have a TM that accepts $A$, and that halts on every input, i.e., it decides $A$, and so $A$ is recursive.

This shows that $\Sigma_1 \cap \Pi_1 \subseteq \Sigma_0$. The other inclusion ($\Sigma_0 \subseteq \Sigma_1 \cap \Pi_1$) follows trivially from the fact that every recursive language is both r.e. and co-r.e.
(b) [12.5 points] Give an explicit, direct, many-one reduction (note: on this particular problem, you do not have to include a correctness proof; just give a correct reduction) from $\overline{HP}$ to $D = \{i \mid L(M_i) \text{ is infinite}\}$.

(Comment: Does this look familiar? It solves one of the two questions from the Diagnostic Quiz you took on the first day of class—in fact, it proves even a bit more than that was asking. On the question on the Diagnostic Quiz, more than 3/4 of the class got 0 points; I’m hoping things will go substantially better here!)

Answer:

$\sigma(i) = j$, where $j$ is the index of a TM $M_j$ that does the following on arbitrary input $y$:

Simulate $M_i$ on $i$ for $|y|$ steps. If it halts and accepts within that time, reject. Otherwise, accept.
**Question 5 [25 points]** Let $A$ and $B$ be as in problem 2. Namely, $A = \{ i \mid L(M_i) = \emptyset \}$ and $B = \{ (j,k) \mid L(M_j) = L(M_k) \}$. Give an explicit, direct, many-one reduction from $\overline{A}$ to $B$ (i.e., $\overline{A} \leq_m B$), and prove (in our 3-part way) that it is indeed is such a reduction.

**Answer:**

The function $\sigma$ will provide such a reduction. Fix an integer $j_0$ such that $L(M_{j_0}) = \Sigma^*$. Let $\sigma(i) = (j_0, k)$ where $j_0$ is as set above, and $k$ is the index of a TM $M_k$ that does the following on arbitrary input $y$:

Run an enumerator (i.e., an enumerating Turing machine) for $L(M_i)$ (note: we can effectively map from $i$ to an enumerating Turing machine for $L(M_i)$). If it enumerates a string, then accept.

$\sigma$ is clearly a recursive function.

If $i \in \overline{A}$, then $L(M_i)$ contains at least one string. Thus an enumerator for $M_i$ will eventually enumerate a string and so $L(M_k) = \Sigma^*$, and so $(j_0, k) \in B$.

If $i \notin \overline{A}$, $L(M_i) = \emptyset$. Thus an enumerator for $L(M_i)$ will never enumerate any strings and so $L(M_k) = \emptyset$. Thus $(j_0, k) \notin B$.

We have that for each $i$ it holds that $i \in \overline{A} \iff \sigma(i) \in B$, and $\sigma$ is a recursive function, so we have that $\overline{A} \leq_m B$. 
**Question 6** [25 points] Prove that there is a set $A$, over the alphabet \{0, 1\}, such that (a) $A$ is an infinite set, (b) $A$ has no infinite recursively enumerable subset, and (c) for each $n \geq 486$ it holds that of the $2^n$ distinct strings of length $n$ at least 2016 of them are elements of $A$. (Note: If in your proof you prove that $A$ will contains at least 2016 distinct strings of length 2, there probably is a problem in your proof.)

**Answer:** (Note: Myself, I most like doing such proofs as a very explicit stage construction, with (usually) $A_0 = \emptyset$ initially, and $A = \bigcup_{i \geq 0} A_i$, and $A_{i+1} \supseteq A_i$, and the $i$th stage ($i \geq 1$) building $A_i$, and with a variable being used to keep track of where we are up to, and so on. But the more terse treatment below is livable here.)

Let $M_1, M_2, \ldots$ be an enumeration of those TMs that accept infinite languages. We will describe a process for constructing an $A$ meeting the above conditions.

Do the following for every $i$ from 1 on: Find the next string $x$ accepted by $M_i$ that is lexicographically greater than all strings yet considered. (Since we have so far considered only a finite number of strings, and $L(M_i)$ is infinite, there will exist such a string.) Exclude $x$ from $A$. Include in $A$ all other strings of the length of $x$ and also all strings shorter than $x$ which had not yet been considered.

This ends the construction.

For every TM $M$ accepting an infinite language, there is a string $x$ such that $x \in L(M)$ and $x \notin A$. Thus $M$ has no infinite r.e. subset.

$A$ contains at least $2^n - 1$ strings of each length $n$, since at most one string is excluded at each length (and all the other strings at that length are put into the set). Thus $A$ is infinite and also $A$ contains at least 2016 strings of each length strictly greater than 10 (there will be 1023 or 1024 strings in $A$ of length 10, but already at length eleven we’ll have 2047 or 2048 strings in $A$, which is certainly at least 2016 strings).

The constructed set $A$ is a set satisfying the required conditions.