COMP330 Assignment 4

Ryan Sowa, ID: 260886668

1. (10 points) Let M_1 and M_2 be two Turing machines. Consider the following Turing machine: On input w:

   - Step 1: Run M_1 on w. If M_1 accepts w, then accept.
   - Step 2: Run M_2 on w. If M_2 accepts w, then accept.

   What is the language of this Turing Machine? Explain.

   Let L_1 be the language of M_1 and let L_2 be the language of M_2.

   If M_1 loops on w, then even if M_2 accepts w, it will not be able to run on w. Let the set of words that M_1 loops on be denoted by $S = s_1, s_2 ... s_n$.

   If M_1 does not accept w or loop on w, then M_2 will be able to run on w.

   Therefore, the language of this Turing Machine is $(L_1 \cup L_2) \setminus S$.

2. (15 points) Let E be an enumerator for a language L with the extra property that it will print the words in an increasing order of lengths. That is it will never print a word w before a shorter word u.

   Prove that L is decidable.

   Proof. Suppose by contradiction that L is undecidable. To print out all the words in L in increasing order of lengths, we must try all combinations (using all the letters in the alphabet), of length 1, length 2, etc. If L accepts that word, then E prints it. Eventually, all the words in L would be printed in increasing order of lengths.

   However, consider the case when we check if w, a word which L loops on (L is undecidable), is in L. Since L will be looping on w, we will never be able to print out the other words. To print out the other words, L must not be undecidable $\Rightarrow$ L is decidable.

   $\square$

3. (15 points) Is the following language decidable?

   $$L = \{\langle M \rangle | M = (\{1, 2, ..., 100\}, \{0, 1\}, \{0, 1, \sqcup\}, \delta, 1, 2, 3) \text{ is a decider}\}.$$

Proof. First, we notice that since M must have 100 states, L must be finite. Therefore, we can construct a list of all Turing Machines of 100 states that are deciders. If the input is of the wrong format, we reject. On input $\langle M \rangle$ to L, if M matches any of the Turing Machines in our list, then accept. Otherwise, reject.

Thus, L is decidable.

$\square$

4. (20 points) In the class we showed that the following language is Turing Recognizable. Prove that it is in fact decidable.

   $L = \{\langle p(x) \rangle |\ p(x)$ is a polynomial with integer coefficients that has an integer root$\}$.

   Proof. First, if our input is in the wrong format (not $p(x)$), we reject. If it is in the right format, since $p(x)$ is a polynomial, to find its roots we can set it to zero and write it in the form

   $$p(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_0 = 0$$

   for $a_i \in \mathbb{Z}$, $a_n \neq 0$. If $a_0 = 0$, then $x = 0$ is a solution, so we analyze when $a_0 \neq 0$.

   Since $a_n, a_0 \neq 0$, we can use the Rational Root Theorem to find the roots of $p(x)$. Since the factors of a_n and a_0 are finite, the (plus or minus) quotient of the factors of a_0 and a_n will also be finite, namely (factors of a_0) * (factors of a_n) * 4 by the Rational Root Theorem. This implies we will be testing a finite number of (integer) numbers, $\alpha_a, ..., \alpha_z$.

   If one of these α_i's gives a root to $p(x)$, we accept. If after testing all α_i's, none of them produced a root, we reject.

   $\square$

5. (20 points) Without using Rice's theorem (See Problem 5.28 of the textbook) prove that the following language is not decidable

   $$L = \{\langle M \rangle |\ L(M) \text{ is finite}\}.$$

   Suppose by contradiction that L is decidable and can be decided by a Turing Machine R. We will attempt to prove that if this is the case, then A_{TM} is also decidable (which is not true).

   On input $\langle M, W \rangle$ to A_{TM}:

   1. Construct the description of the following Turing Machine N:

On input x, use U to run M on w. If it accepts, then accept. Else, reject.

The language of N is

$$\begin{cases} \Sigma^* \; M \text{ accepts } w \\ \phi \; M \text{ does not accept } w. \end{cases}$$

2. Use U to run R on $\langle N \rangle$.

3. (We know that R will accept ϕ since it is finite and will not accept Σ^* since it is infinite.) If R accepts, $(L(N) = \phi)$ then reject $\langle M, W \rangle$ for A_{TM}. If R rejects, $(L(N) = \Sigma^*)$ then accept $\langle M, W \rangle$ for A_{TM}.

Since we have proved that if L is decidable, A_{TM} is also decidable, this is a contradiction. $\Rightarrow\Leftarrow$ Thus, L is undecidable.

6. (20 points) You are allowed to use Rice's theorem (See Problem 5.28 of the textbook) to answer this question. For each of the following three languages, either prove that they are decidable, or prove that they are undecidable.

   (a)
   $$L_r = \{\langle M \rangle | \text{ L(M) is a regular language}\}.$$

   Proof. Condition 1: Trivial property. Since there are Turing Machines in L_r and Turing Machines not in L_r, L_r is a nontrivial property.

   Condition 2: Since $M_1 = \{a^n b^n\} \notin L_r$ and $M_2 = \{x | \; x \text{ ends in “11”}\} \in L_r$, by Rice's Theorem, L_r is undecidable. □

   (b)
   $$L_{TR} = \{\langle M \rangle | \text{ L(M) is a Turing recognizable language}\}.$$

   Proof. Condition 1: Trivial property. Since all Turing Machines are clearly in L_{TR}, L_{TR} is a trivial property and is (by Rice's Theorem) decidable. □

   (c)
   $$L_d = \{\langle M \rangle | \text{ L(M) is a decidable language}\}.$$

   Proof. Condition 1: Trivial property. Since there are Turing Machines in L_d and Turing Machines not in L_d, L_d is a nontrivial property.

Condition 2: Since $M_1 = A_{TM} \notin L_d$ and $M_2 = \{\langle C, w\rangle |\ C$ is a CFG and $w \in L(C)\}$ $\in L_d$, by Rice's Theorem, L_d is undecidable.

$\square$