COMP330 Assignment 5

Ryan Sowa, ID: 260886668

1. (5 points) Prove that every Turing Recognizable Language L satisfies $L \leq_m A_{TM}$.

   Proof. A_{TM} is defined as $A_{TM} = \{\langle M, w\rangle \mid \text{M accepts w}\}$. Since L is Turing Recognizable, let T be the Turing Machine for L such that $L(T) = L$.

   On input x to T, run A_{TM} on $\langle T, x\rangle$. Since

   $$x \in L \Leftrightarrow \langle T, x\rangle \in A_{TM}.$$

   is true, we have shown that $L \leq_m A_{TM}$.

   □

2. (10 points) Prove that for any two languages A and B, there exists a language J such that $A \leq_T J$ and $B \leq_T J$.

   Proof. Construct J as follows: 1. The language of J is $A \cup B$ 2. All the strings in J that are also in A are colored red 3. All the strings in J that are also in B are colored blue.

   $A \leq_T J$:

   On every input x to A, we color x red and use M^J to see if $x \in J$. If so, then accept. Otherwise, reject.

   $B \leq_T J$:

   On every input x to B, we color x blue and use M^J to see if $x \in J$. If so, then accept. Otherwise, reject.

   □

3. (10 points) Prove that for any language A, there exists a language J such that $A \leq_T J$ and $J \not\leq_T A$.

   Proof. Let

   $$M_{oracle} = \{\langle M\rangle \mid M \text{ is a Turing Machine which has an oracle for } A\}$$

and let

$$J = \{\langle M, w\rangle \mid M \in M_{oracle}\ ,\ M \text{ accepts } w\}.$$

$A \leq_T J$:

On input w to A, construct a Turing Machine N for A. Then, use the oracle as described:

If $\langle N, w\rangle \in J$, then accept.
If $\langle N, w\rangle \notin J$, then reject.

$J \not\leq_T A$:

We use the diagonalization argument to show this. Suppose by contradiction that there exists a J-oracle M^A that decides J. This means we can decide J relative to A, and we can therefore build a table based on how each Turing Machine in J, M_i behaves on each input, w_i. However, we can construct

$$A_{diagonal} = \{w_i \mid i \in N \text{ and } M_i \text{ does not accept } w_i\}$$

from this table. Since M^A cannot decide $A_{diagonal}$, it is impossible to decide J relative to A, so $J \not\leq_T A$.

□

4. (15 points) Let $\Gamma = \{0, 1, \sqcup\}$ be the tape alphabet for all Turing Machines in this problem. Define the busy beaver function $\mathbf{BB} : \mathbb{N} \to \mathbb{N}$ as follows. For each value of k, consider all k-state Turing Machines that halt when started with a blank tape. Let **BB**(k) be the maximum number of 1's that remain on the tape among all of those machines. Prove that **BB** is not a computable function.

Proof. Suppose by contradiction that **BB** is computable. Then, we will show that A_{TM} is decidable (which is not true).

On input $\langle M, w\rangle$ to A_{TM}, build a Turing Machine N which constructs a Turing Machine T. T:

1. Starts on an empty tape and writes out w
2. Simulates M on w
3. If M ever halts (accepts or rejects), writes 1's in every cell M ever used

Let the number of states in T be denoted by s. N can compute the maximum number of 1's that remain on T's tape as $n = \mathbf{BB}(s)$.

N marks the n'th cell from the left of the tape as the "must halt" cell and then runs T. If T goes past the n'th cell, N rejects (no string which halts goes to this cell). If M loops on w, we will be able to detect a cycle within our n cells, so N rejects. Otherwise, if T accepts, then N accepts. If T rejects, then N rejects.

Run N on $\langle M, w\rangle$. If N accepts, then accept. Else, reject.

Since, we have proved A_{TM} is decidable (a contradiction) if **BB**(k) is computable, **BB**(k) must not be computable.

$\square$

5. (15 points) Let $f : \mathbb{N} \to \mathbb{N}$ be defined as

$$f(x) = \begin{cases} 3x+1 & x \text{ is odd} \\ x/2 & x \text{ is even} \end{cases}$$

For every natural number x, if you start with x and iterate f, you obtain a sequence

$$x, f(x), f(f(x)), f(f(f(x))), \ldots.$$

Stop if you every hit 1. For example, if $x = 17$, we get the sequence

$$17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.$$

It has been checked by computers that if we start with any number $x < 10^{17}$, we will eventually hit 1 and terminate, but it is unknown whether this is true for every integer x. This is known as Collatz's conjecture. Suppose that $HALT_{TM}$ was decidable, and R was a TM that was deciding $HALT_{TM}$. Construct a Turing Machine M based on R such that it accepts ε if Collatz's conjecture is true (all numbers x hit 1 eventually), and rejects if Collatz's conjecture is false (there is some x that does not hit 1).

Proof. Description of M:

On input ε, construct the following Turing Machine N:

For $j = 1, 2...$

On input $x = j$, for $i = 0, 1, 2...$ iterate f on x i times. End this (inner) for loop if we get a final result of 1.

Accept if all iterations of j ended in a final result of 1.

Use R as follows:

If $\langle N, \varepsilon\rangle \in HALT_{TM}$, then accept ($N$ halted, so all numbers hit 1 eventually).

If $\langle N, \varepsilon\rangle \notin HALT_{TM}$, there exists some number which did not reach 1 (and therefore, N did not halt), so reject.

$\square$

6. (15 points) Let

$X = \{\langle M, w\rangle \mid M$ is a single-tape TM that never modifies the w part of the tape, and it accepts $w\}$.

Is X decidable? Prove your answer.

Proof. Suppose by contradiction that X is decidable. Then, we will prove that A_{TM} is also decidable (which is impossible).

Let R be the Turing Machine which decides X.

On input $\langle M, w\rangle$ to A_{TM}, construct a Turing Machine N which does the following:

On any input, add a "\$" after the input part. Then, write w after the \$. Run M on w. If M accepts, then write something on the input part of the tape and accept. Else, reject.

Now, run R on $\langle N, w\rangle$. If R accepts, this mean we wrote on our input part of our tape and M accepted w, so accept. If R rejects, this mean we never were able to write on our input part of the tape since M never accepted w, so reject.

Since we have shown that if X is decidable, then A_{TM} is also decidable, X must be undecidable.

□

7. (15 points) Prove that a language L is Turing Recognizable if and only if it can be expressed as

$$L = \{x \mid \exists y \text{ such that } \langle x, y\rangle \in R\ \}$$

where R is a decidable language. You need to prove that every language of this form is Turing Recognizable, and that every Turing Recognizable language can be described as above for some decidable language R.

Proof. Suppose L is Turing Recognizable. Let T be the Turing Machine that recognizes L and let

$$R = \{\langle x, y\rangle \ \mid\ T \text{ accepts } x \text{ after no more than } |y| \text{ steps}\}.$$

R is decidable since it can be decided as follows:
On input $\langle x, y\rangle$ to R, if T accepts x after no more than y steps, then accept. Else, reject.

If some input x is in L, then it must be accepted by T after some number of steps. In other words, $\langle x, y\rangle$ must be in R. Therefore, $L = \{x \mid \exists y \text{ such that } \langle x, y\rangle \in R\ \}$ must hold.

On the other hand, suppose $L = \{x \mid \exists y \text{ such that } \langle x, y\rangle \in R\ \}$, where R is a decidable language. We can construct a Turing Machine which recognizes L as follows: for any input x, test every single y to see if $\langle x, y\rangle \in R$. If the condition is satisfied, accept. Otherwise, keep searching for y (looping is fine for a Turing Recognizable language).

□

8. (15 points) Determine whether the following language is decidable or undecidable:

$$X = \{\langle M \rangle \mid \text{On every input } w,\ M \text{ eventually leaves the start state}\}.$$

If input is of the wrong form, reject. On input $\langle M \rangle$, examine all the transition states (finite).

If there exists any transition from the start state to the reject or accept state, then reject (would never leave start state).

If there exists any transition from the start state to itself, then reject (could loop at the start state).

Else, accept (would have to leave start state eventually on all inputs).