**CS 420 Spring 2019**
**Homework 10 Solutions**

1. (a) $REJECT_{TM}$ is defined as $\{\langle M, w\rangle | M$ is a Turing machine, and $M$ rejects $w\}$. Prove that $REJECT_{TM}$ is Turing recognizable.

   **Solution:** $REJECT_{TM}$ is recognized by the following Turing machine $V$:

   $V =$ " On input $\langle M, w\rangle$

   1. Run $M$ on $w$.
   2. If $M$ rejects, *accept*. If $M$ accepts, *reject*."

   (b) Show that $REJECT_{TM}$ is undecidable using diagonalization. Your proof should be similar to, but not the same as, the proof that $A_{TM}$ is undecidable.

   **Solution:** We assume that $REJECT_{TM}$ is decidable and obtain a contradiction. Suppose that the Turing machine $H$ decides $REJECT_{TM}$. This means that

$$H(\langle M, w\rangle) = \begin{cases} \textit{reject} & \text{if } M \text{ accepts } w \\ \textit{accept} & \text{if } M \text{ rejects } w \\ \textit{reject} & \text{if } M \text{ loops on } w \end{cases}$$

   Using $H$, we define another Turing machine $D$

   $D =$"On input $\langle M\rangle$ where $M$ is a Turing machine

   1. Run $H$ on $\langle M, \langle M\rangle\rangle$.
   2. If $H$ accepts, *accept*. If $H$ rejects, *reject*."

   We have

$$D(\langle M\rangle) = \begin{cases} \textit{reject} & \text{if } M \text{ accepts } \langle M\rangle \\ \textit{accept} & \text{if } M \text{ rejects } \langle M\rangle \\ \textit{reject} & \text{if } M \text{ loops on } \langle M\rangle \end{cases}$$

   Applying this to $M = D$, we get

$$D(\langle D\rangle) = \begin{cases} \textit{reject} & \text{if } D \text{ accepts } \langle D\rangle \\ \textit{accept} & \text{if } D \text{ rejects } \langle D\rangle \\ \textit{reject} & \text{if } D \text{ loops on } \langle D\rangle \end{cases}$$

   No matter what $D$ does on $\langle D\rangle$, we get a contradiction, so $D$ can't exist, which means that $H$ can't exist and $REJECT_{TM}$ is undecidable.

   (c) Give a second proof that $REJECT_{TM}$ is undecidable by reducing $A_{TM}$ to $REJECT_{TM}$.

   [This will involve some creativity because the technique we used to reduce $A_{TM}$ to $HALT_{TM}$ will not work here.]

   **Solution:** Suppose that the Turing machine $R$ decides $REJECT_{TM}$. Then we define a Turing machine $S$ that decides $A_{TM}$ as follows:

   $S =$"On input $\langle M, w\rangle$

1. Construct a TM $M'$ from $M$ by reversing the accept and reject states.
2. Run $R$ on $\langle M', w\rangle$.
3. If $R$ accepts, *accept.* If $R$ rejects, *reject.*"

Note that $\langle M, w\rangle \in A_{TM}$ if and only if $M$ accepts $w$ if and only if $M'$ rejects $w$ if and only if $\langle M', w\rangle \in REJECT_{TM}$ if and only if $R$ accepts $\langle M, w\rangle$ if and only if $S$ accepts $\langle M, w\rangle$, so $S$ decides $A_{TM}$. Since $A_{TM}$ is undecidable, no such Turing machine $S$ exists, so $R$ cannot exist and $REJECT_{TM}$ is undecidable.

2. The proof is the same as the one given in Theorem 5.3, except that in step 1 of the instructions for $M_2$, "has the form $0^n1^n$" is replaced by "has the form $0^n1^n2^n$". Now if $M$ accepts $w$, then $L(M_2)$ is the context-free language $\Sigma^*$ and if $M$ does not accept $w$, then $L(M_2)$ is the non-context-free language $\{0^n1^n2^n | n \geq 0\}$.

3. Let $NONREGULAR_{TM} = \{\langle M\rangle | M$ is a Turing machine, and $L(M)$ is not a regular language$\}$. Suppose that you want to reduce $A_{TM}$ to $NONREGULAR_{TM}$ by transforming $\langle M, w\rangle$ to $\langle M_2\rangle$. (So if $\langle M, w\rangle$ is in $A_{TM}$, then $\langle M_2\rangle$ is in $NONREGULAR_{TM}$, and if $\langle M, w\rangle$ is not in $A_{TM}$, then $\langle M_2\rangle$ is not in $NONREGULAR_{TM}$.)

   (a) Fill in the blanks in the following two statements in a way that states what you have to do to make the reduction work. Make your statements as general as possible. (In both cases you will be writing down something about the behavior of the Turing machine $M_2$.)

   - If $M$ accepts $w$, then
     $L(M_2)$ is not regular.
   - If $M$ does not accept $w$, then
     $L(M_2)$ is regular.

   (b) Give the definition of the desired Turing machine $M_2$, given $M$ and $w$.

   $M_2$ = "On input $x$

   1. If $x$ does not have the form $a^nb^n$ for some $n$, *reject.*
   2. If $x = a^nb^n$ for some $n \geq 0$, run $M$ on $w$.
   3. If $M$ accepts $w$, *accept.* If $M$ rejects $w$, *reject.*

   If $M$ accepts $w$, then $L(M_2) = \{a^nb^n | n \geq 0\}$, so $L(M_2)$ is not regular. If $M$ does not accept $w$, then $L(M_2) = \emptyset$, so $L(M_2)$ is regular.

4. **Problem 5.9**

   We show that $T$ is undecidable by reducing $A_{TM}$ to $T$. Suppose that $R$ is a TM that decides $T$. We will give a TM $S$ that decides $A_{TM}$. Since

$A_{TM}$ is not decidable, this will show that no such TM $R$ can exist, so $T$ is not decidable.

$S$ will have this form:

$S =$ "On input $\langle M, w \rangle$ where $M$ is a Turing machine and $w$ is an input:

1. Produce the TM $M_1$.
2. Run the TM $R$ that decides $T$ on $\langle M_1 \rangle$.
3. If $R$ accepts, then *accept*. If $R$ rejects, then *reject*."

To complete the proof, we have to show how, given $\langle M, w \rangle$, $S$ can produce the description of a TM $M_1$ such that if $M$ accepts $w$, then $\langle M_1 \rangle \in T$, i.e., whenever $M_1$ accepts a string $w$, then it also accepts $w^R$, and if $M$ does not accept $w$, then $\langle M_1 \rangle \notin T$, i.e., there is some string $w$ such that $M_1$ accepts $w$ but does not accept $w^R$. Here is a description of $M_1$.

$M_1 =$ "On input $x$,

1. If $x = 01$, then accept $x$.
2. If $x \neq 01$, then run $M$ on input $w$ and *accept* if $M$ accepts."

If $M$ accepts $w$, then $M_1$ accepts every string, so $\langle M_1 \rangle$ belongs to $T$. If $M$ does not accept $w$, then the only string accepted by $M_1$ is 01, so $\langle M_1 \rangle$ does not belong to $T$.

5. **Problem 5.14**

Let $S = \{\langle M, w \rangle | M$ is a Turing machine, $w$ is an input and at some point in its computation on $w$, $M$ moves its head left when it is reading the leftmost cell$\}$. We show that $S$ is undecidable by reducing $A_{TM}$ to $S$.

Given a TM $M$, we define a new TM $M_1$ that works as follows. Given input $w$, $M_1$ simulates $M$ on $w$, but $M_1$ keeps a special mark on the first tape cell. When $M$ moves left from the first tape cell, $M_1$ instead moves right and then back to the left, ending up in the same state $M$ would be in. If $M$ ever accepts $w$, then $M_1$ moves left to the first cell and then moves left once more. If $M$ rejects $w$, then $M_1$ just rejects without moving left. Thus $M$ accepts $w$ if and only if $M_1$ at some point in its computation on input $w$ moves left from the leftmost cell. Because a Turing machine can carry out the construction of $M_1$, given $M$, this is a reduction from $A_{TM}$ to $S$ and shows that $S$ is undecidable.

More formally, suppose that $R$ is a TM that decides $S$. Then the following TM $N$ would decide $A_{TM}$, which is impossible.

$N =$ "On input $\langle M, w \rangle$ where $M$ is a Turing machine and $w$ is an input:

1. Produce the TM $M_1$.
2. Run the TM $R$ that decides $S$ on $\langle M_1, w \rangle$.
3. If $R$ accepts, then *accept*. If $R$ rejects, then *reject*."

6. **Problem 5.15**

   The corresponding language is $MOVELEFT_{TM} = \{\langle M, w\rangle | M$ is a Turing machine and on input $w$ $M$ attempts to move its head left at some point in the computation$\}$.

   A Turing machine to decide $MOVELEFT_{TM}$ implements the following algorithm. Given $M$ and $w$, simulate $M$ on $w$ until either a) $M$ attempts to move left, b) $M$ halts without ever trying to move left, or c) $M$ moves right beyond the original input and then repeats some state without ever trying to move left. One of these three things has to happen eventually because if $M$ never tries to move left and never halts, it will always move right and eventually move past the original input. Since there are only finitely many states, eventually a state will be repeated after $M$ has gone past its original input. If a) happens, then $\langle M, w\rangle \in MOVELEFT_{TM}$. If b) happens, then $\langle M, w\rangle \notin MOVELEFT_{TM}$. If c) happens, then we also have $\langle M, w\rangle \notin MOVELEFT_{TM}$ because if $M$ goes from a configuration $up\sqcup$ to a configuration $uvp\sqcup$ without ever moving left, then $M$'s computation will continue as $uvvp\sqcup$, $uvvvp\sqcup$, $\ldots$ without ever trying to move left. Since a Turing machine can implement this algorithm, $MOVELEFT_{TM}$ is decidable.

7. **Problem 27**

   Let $EQ_{2DIM-DFA} = \{\langle A, B\rangle | A$ and $B$ are 2DIM-DFA and $L(A) = L(B)\}$. We show that $EQ_{2DIM-DFA}$ is undecidable.

   Let $E_{2DIM-DFA} = \{\langle A\rangle | A$ is a 2DIM-DFA and $L(A) = \emptyset\}$. $E_{2DIM-DFA}$ is $m$-reducible to $EQ_{2DIM-DFA}$ in the same way that $E_{TM}$ is $m$-reducible to $EQ_{TM}$, so if we show that $E_{2DIM-DFA}$ is undecidable, this will show that $EQ_{2DIM-DFA}$ is undecidable.

   To show that $E_{2DIM-DFA}$ is undecidable, we will $m$-reduce $A_{TM}$ to $\overline{E_{2DIM-DFA}}$ by a mapping which takes $\langle M, w\rangle$ to $\langle B\rangle$. This means that if $M$ accepts $w$, then we want $L(B)$ to be non-empty and if $M$ does not accept $w$, we want $L(B)$ to be empty. We accomplish this by making $L(B)$ be the set of accepting computation histories of $M$ on $w$. A computation history $C_1, \ldots, C_k$ is presented to $B$ as a rectangle with $C_1$ in the first row, $C_2$ in the second row, etc. (Short configurations are padded with blanks on the right end.) Given an input rectangle, $B$ checks that the first row is the initial configuration of $M$ on $w$, that the last row is an accepting configuration, and that each row follows from the previous row by the rules of $M$. ($B$ can't write, but because the configurations are one on top of the other, $B$ can still check that one configuration $C_{i+1}$ follows correctly from $C_i$ – the only entries in $C_{i+1}$ that could be different from the entries in $C_i$ are the ones within one cell of where the state is in $C_i$.) If the rectangle passes all these tests, then $B$ accepts, else $B$ rejects.

   If $M$ accepts $w$, then there is an accepting configuration history of $M$ on $w$ and $L(B) \neq \emptyset$. If $M$ does not accept $w$, then there is no accepting computation history of $M$ on $w$ and $L(B) = \emptyset$. Thus, we have $m$-reduced $A_{TM}$ to $\overline{E_{2DIM-DFA}}$ and $E_{2DIM-DFA}$ is undecidable. It follows that $EQ_{2DIM-DFA}$ is undecidable.