Problem Set 1 Solutions

Problem 1. [40 points] If \( f : \Sigma^* \to \Sigma^* \) is a function we define the image of \( f \) as
\[
\text{Img}(f) = \{ f(w) : w \in \Sigma^* \}.
\]
Let \( L \) be a non-empty language. Prove that \( L \) is r.e. if and only if there exists a computable function \( f : \Sigma^* \to \Sigma^* \) such that \( L = \text{Img}(f) \).

The two directions of the “if and only if” are proved separately.

Claim 1: Suppose there exists a computable function \( f : \Sigma^* \to \Sigma^* \) such that \( L = \text{Img}(f) \). Then \( L \) is r.e..

Proof: Let \( M_f \) be a TM that computes the function \( f \). We specify a verifier \( V \) for \( L \):

Verifier \( V(x, w) \)
- If \( M_f(w) = x \) then accept else reject

The “certificate” for the membership of \( x \) in \( L \) is a pre-image of \( x \) under \( f \), namely a point \( w \) such that \( f(w) = x \). This exists if and only if \( x \in L \) because \( L = \text{Img}(f) \).

Claim 2: Suppose \( L \) is r.e.. Then there exists a computable function \( f : \Sigma^* \to \Sigma^* \) such that \( L = \text{Img}(f) \).

Proof: Let \( M \) be a TM that recognizes \( L \). Since \( L \) is non-empty we may fix a point \( x_0 \in L \). Let \( M' \) be the following TM that takes as input a string \( w \):

\[ M'(w) \]
- If \( w \) is not the encoding of a pair of strings then return \( x_0 \)
- Else
  - Let \( x, y \) be the strings such that \( w = \langle x, y \rangle \)
  - Run \( M(x) \) for \(|y|\) steps
  - If it accepts then return \( x \) else return \( x_0 \)

We define the function \( f \) by \( f(w) = M'(w) \) for all \( w \in \Sigma^* \). (This function is well-defined since \( M'(w) \) halts with some output for all \( w \in \Sigma^* \), and furthermore is computable by definition.)

Now, for correctness, the claim is that \( \text{Img}(f) = L \) where \( f \) is defined above. To prove this, we show two things: that \( \text{Img}(f) \subseteq L \) and that \( L \subseteq \text{Img}(f) \).
To see that $L \subseteq \text{Img}(f)$ let $x$ be any string in $L$. We need to show that $x \in \text{Img}(f)$, meaning there is some $w$ such that $M'(w) = x$. Since $x \in L$ we know that $M(x)$ accepts, and we can let $y$ be such that $M(x)$ accepts in $|y|$ steps. Now set $w = \langle x, y \rangle$, and we have $M'(w) = f(w) = x$.

To see that $\text{Img}(f) \subseteq L$ let $x$ be any string in $\text{Img}(f)$. We need to show that $x \in L$. Since $x \in \text{Img}(f)$, there is some $w$ such that $M'(w) = x$. By definition of $M'$ it must be that either $w = \langle x, y \rangle$ for some $y$ such that $M(x)$ accepts in $|y|$ steps, or $x = x_0$. In either case, $x \in L$.

Note the first claim did not use the assumption that $L \neq \emptyset$, but the second claim did use it. Indeed, the second claim is not true without this assumption.

Problem 2. [30 points] Prove that the following language is decidable:

$$A = \{ \langle M \rangle : \text{The head of TM } M \text{ stays within the first 2011 tape squares in } M's \text{ computation on input } \varepsilon \}.$$

To show that $A$ is decidable, we need to specify a TM $M_A$ that on input $\langle M \rangle$ accepts if $\langle M \rangle \in A$ and rejects otherwise. Note that $M_A$ must halt on all inputs.

The natural strategy for $M_A$ to attempt is to start running $M$ on input $\varepsilon$, keeping track of the number of tape squares that $M$ is using. If the computation $M(\varepsilon)$ ever hits the 2012-th tape square, then $M_A$ can reject. If $M(\varepsilon)$ halts having stayed within the first 2011 squares then $M_A$ can accept. But what if $M(\varepsilon)$ keeps going, staying all the while within the first 2011 squares? The problem is: if $M_A$ stops, thinking $M(\varepsilon)$ will never hit square 2012, maybe it would have had it run longer.

To address this, recall that a configuration of TM $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ is a triple $(u, q, v)$ where $q$ is the current state, $u$ is the string to the left of the head, excluding the scanned symbol, and $v$ is the string to the right of the head, including the scanned symbol and up to the last non-blank. We claim that the following algorithm $M_A$ decides $A$:

Algorithm $M_A(\langle M \rangle)$

$\mathcal{C} \leftarrow \emptyset$

Repeat

Run $M(\varepsilon)$ for one more step and let $C$ be the resulting configuration

If the 2012-th square has been hit then reject

If the configuration $C$ is a halting one then accept

If $C \in \mathcal{C}$ then accept

$\mathcal{C} \leftarrow \mathcal{C} \cup \{ C \}$

Now we want to argue that this algorithm is correct.

Let $\mathcal{C}(M)$ be the set of all possible configurations for the computation $M(\varepsilon)$ in which the used part of the tape consists of at most 2011 squares, meaning $|uv| \leq 2011$. Let $N(M)$ denote the size of the set $\mathcal{C}(M)$. Then

$$N(M) \leq |\Gamma|^{2012} \cdot |Q| \cdot 2012.$$

Now, let $C_1, C_2, \ldots$ be the sequence of configurations of $M$ running on input $\varepsilon$. If $M$ stays within 2011 squares then all these configurations are from $\mathcal{C}$. The key observation is that if it stays
within 2011 squares for more than $N(M)$ moves, it must at some point return to an already used configuration; that is, $C_j = C_i$ for some $i < j \leq N + 1$. But if so, its future is determined: after reaching $C_j$ it will keep repeating the sequence $C_{i+1}, \ldots, C_j$. That is, it is simply in an infinite loop inside the 2011 squares. So it will never go the 2012-th square, meaning $M_A$ can accept. This shows that the above algorithm always halts in at most $N(M)$ simulation steps, and takes the correct decision.