Solutions of homework 5

1. *Sipser problem 4.19.*

   SOLUTION OUTLINE: (20 POINTS)

   Here's an algorithm for the language S: on input $\langle M \rangle$, reject if $\langle M \rangle$ is not a valid encoding of a DFA. Otherwise, write $L = L(M)$ for ease of notation. First, construct a DFA $M^{\mathcal{R}}$ for $L^{\mathcal{R}}$ by converting the NFA for $L^{\mathcal{R}}$ (from problem set 1) into a DFA. Next, construct a DFA A for $L \triangle L^{\mathcal{R}}$ by running M and $M^{\mathcal{R}}$ in parallel and accepting iff exactly one of the two DFAs accepts.[1] Now, using the decider for E_{DFA}, accept if $A \in E_{\mathsf{DFA}}$ and reject otherwise.

   To show correctness, observe that $A \in E_{\mathsf{DFA}}$ iff for all $w \in \Sigma^*$, either $w \in L$ and $w \in L^{\mathcal{R}}$, or $w \notin L$ and $w \notin L^{\mathcal{R}}$. The latter is equivalent to M accepting $w^{\mathcal{R}}$ whenever it accepts w.

2. *Sipser problem 5.14.*

   SOLUTION OUTLINE: (20 POINTS)

   We may formalize the problem as the following language: $B = \{\langle M, w\rangle \mid M$ is a TM that on input w ever attempts to move its head left when its head is on the left-most tape cell.$\}$. To see that B is undecidable, assume on the contrary that there exists some TM R that decides B, and we use R to construct a TM S that decides A_{TM}:

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

   1. Construct TM M' on input w, copies w onto the tape one position right, and writes a special symbol $\square$ on the first position. Then, M' simulates M on input w starting from the second tape position, with two changes. First, if the head reads the symbol $\square$, it moves right, and stays in the same state. Second, if M enters an accept state, M' enters a special state where the head just keeps moving to the left (past the left-most tape cell).
   2. Run R on input $\langle M', w\rangle$.
   3. Accept if R accepts, and reject if R rejects."

3. (a) *Prove that $\overline{E_{\mathsf{TM}}}$ is Turing-recognizable.*

   (b) *Prove that A_{TM} is not mapping reducible to E_{TM}.*

   SOLUTION OUTLINE: (15,10 POINTS)

   (a) We construct a Turing machine that recognizes $\overline{E_{\mathsf{TM}}}$ as in the construction of an enumerator for a Turing-recognizable language. On input $\langle M \rangle$, for $i = 1, 2, \ldots$, simulate M on all strings of length at most i for i steps, and accept if M accepts any of these strings.

   Note that it follows from (a) and E_{TM} being undecidable that E_{TM} is not Turing-recognizable.

[1] In particular, $Q_A = Q_M \times Q_{M^{\mathcal{R}}}$, $\delta_A((q_1, q_2), \sigma) = (\delta_M(q_1, \sigma), \delta_{M^{\mathcal{R}}}(q_2, \sigma))$ and $F_A = F_M \times \overline{F_{M^{\mathcal{R}}}} \cup \overline{F_M} \times F_{M^{\mathcal{R}}}$.

(b) Suppose on the contrary that A_{TM} is mapping reducible to E_{TM}. Then, the same reduction shows that $\overline{A_{\mathsf{TM}}}$ is mapping reducible to $\overline{E_{\mathsf{TM}}}$. Since $\overline{E_{\mathsf{TM}}}$ is Turing-recognizable, this means that $\overline{A_{\mathsf{TM}}}$ is also Turing-recognizable (using Theorem 5.2 in the text), a contradiction (to Corollary 4.17).

Alternatively, we could use Corollary 5.23 to derive a contradiction to (a).

4. *For each of the following languages, give a proof that it is undecidable or describe an algorithm to decide it.*

   (a) $L_1 = \{\langle M \rangle \mid M$ *is a Turing machine that rejects all inputs of even length*$\}$.

   (b) $L_2 = \{\langle M \rangle \mid M$ *is a Turing machine that halts on an empty input*$\}$.

   (c) $L_3 = \{\langle M \rangle \mid$ *there is some input* $x \in \{0,1\}^*$ *such that* M *accepts* x *in less than 100 steps*$\}$.

Solution Outline: (10,10,15 points)

(a) L_1 is undecidable. To see this, assume on the contrary that there exists some TM R_1 that decides L_1, and we use R_1 to construct a TM S_1 that decides A_{TM}:

$S_1 =$ "On input $\langle M, w \rangle$:

1. Construct TM M_1 that on input x, accept if $|x|$ is odd. If $|x|$ is even, it simulates M on input w. If M accepts w, M_1 enters the reject state. If M rejects w, M_1 enters the accept state. If M loops, M_1 also loops.
2. Run R_1 on input $\langle M_1 \rangle$.
3. Accept if R_1 accepts, and reject if R_1 rejects."

Observe that if M accepts w, then M_1 is a Turing machine that rejects all inputs of even length. If M rejects or loops on input w, then M_1 is a Turing machine that for each input of even length, either loops or accepts.

(b) L_2 is undecidable. To see this, assume on the contrary that there exists some TM R_2 that decides L_2, and we use R_2 to construct a TM S_2 that decides A_{TM}:

$S_2 =$ "On input $\langle M, w \rangle$:

1. Construct TM M_2 that ignores its input and simulates M on input w and accept (and halt) if M does. If M rejects w, M_2 keeps moving right upon reading any input (thereby looping).
2. Run R_2 on input $\langle M_2 \rangle$.
3. Accept if R_2 accepts, and reject if R_2 rejects."

Observe that if M accepts w, then M_2 is a Turing machine that halts on an empty input. If M rejects or loops on input w, then M_2 is a Turing machine that loops on an empty input.

(c) L_3 is decidable. First, observe that if $\langle M \rangle \in L_3$, then there exists some string x of length at most 100 such that M accepts x in less than 100 steps. This is because M cannot read beyond the 100th position of its input in less than 100 steps. Therefore, to check whether an input $\langle M \rangle$ is in L_3, it suffices to simulate M over all strings of length at most 100 for at most 99 steps, and accept if M accepts one of these strings, and reject otherwise.