CS5371 Theory of Computation

Homework 3 (Solution)

1. Show that single-tape TMs that cannot write on the portion of the tape containing the input string recognize only regular languages.

   Answer: Let $M = (Q, \Sigma, \Gamma, q_0, q_{accept}, q_{reject})$ be a single-tape TM that cannot write on the input portion of the tape. A typical case when M works on an input string x is as follows: the tape head will stay in the input portion for some time, and then enter the non-input portion (i.e., the portion of the tape on the right of the $|x|^{th}$ cells) and stay there for some time, then go back to the input portion, and stay there for some time, and then enter the non-input portion, and so on. We call the event that the tape head switches from input portion to non-input portion an *out* event, and the event that the tape head switches from non-input portion to input-portion an *in* event.

   Let first_x denote the state that M is in just after its first "out" event (i.e., the state of M when it first enters the non-input portion). In case M never enters the non-input portion, we assign $\mathit{first}_x = q_{accept}$ if M accepts x, and assign $\text{first}_x = q_{reject}$ if M does not accept x. Next, we define a *characteristic function* f_x such that for any $q \in Q$, $f_x(q) = q'$ implies that if M is at state q and about to perform an "in" event, the next "out" event will change M in state q'; in case M never enters the non-input portion again, we assign $f_x(q) = q_{accept}$ if M enters the accept state inside the input portion, and q_{reject} otherwise.

   It is easy to check that if for two strings x and y, if $\mathit{first}_x = \mathit{first}_y$ and for all q, $f_x(q) = f_y(q)$, we have x and y are indistinguishable by M. (That is, M accepts xz if and only if M accepts yz.) As there are finite choices of first_x and f_x (precisely, $|Q|^{|Q|+1}$ such choices), the number of distinguishable strings are finite. By Myhill-Nerode theorem, the language recognized by M is regular.

2. Let A be a Turing-recognizable language consisting of descriptions of Turing machines, $\{\langle M_1\rangle, \langle M_2\rangle, \ldots\}$, where every M_i is a decider. Prove that some decidable language D is not decided by any decider M_i whose description appears in A.[†] (Hint: You may find it helpful to consider an enumerator for A, and re-visit the diagonalization technique.)

   Answer: Since A is Turing-recognizable, there exists an enumerator E that enumerates it. In particular, we let $\langle M_i\rangle$ be the i^{th} output of E (note: $\langle M_i\rangle$ may not be distinct).

   Let $s_1, s_2, s_3 \ldots$ be the list of all possible strings in $\{\mathtt{0}, \mathtt{1}\}^*$. Now, we define a TM D as follows:

   $D =$ "On input w:

   1. If $w \notin \{\mathtt{0}, \mathtt{1}\}^*$, **reject**.
   2. Else, w is equal to s_i for a specific i.
   3. Use E to enumerate $\langle M_1\rangle, \langle M_2\rangle, \ldots$ until $\langle M_i\rangle$.
   4. Run M_i on input w.
   5. If M_i accepts, **reject**. Otherwise, **accept**."

[†] The question seems strange at the first glance. In fact, it is asking you to prove that the language consisting of *all* descriptions of Turing deciders is not Turing-recognizable.

Clearly, D is a decider (why??). However, D is different from any M_i (why??), so that $\langle D \rangle$ is not in A.

3. Let $E = \{\langle M \rangle \mid M$ is a DFA that accepts some string with more `1`s than `0`s$\}$. Show that E is decidable. (Hint: Theorems about CFLs are helpful here.)

   Answer: Let $A = \{x \mid x$ has more `1`s than `0`s$\}$. The language A is context-free, as we can easily construct a PDA to recognize A. Now, we construct the TM M below to decide E as follows:

   $M =$ "On input $\langle M \rangle$ where M is a DFA:

   1. Construct $B = A \cap L(M)$. Note that B is CFL, since $L(M)$ is regular and A is CFL.
   2. Test whether B is empty.
   3. If yes, **reject**. Otherwise, **accept**.

4. Let C be a language. Prove that C is Turing-recognizable if and only if a decidable language D exists such that $C = \{x \mid \exists y(\langle x, y \rangle \in D)\}$.

   Answer: If D exists, we can construct a TM M such that we search each possible string y, and testing whether $\langle x, y \rangle \in D$. If such y exists, **accept**. Such a machine M will accept any string in C in finite steps, so C is Turing-recognizable.

   If C is recognized by some TM M, we define $D = \{\langle x, y \rangle \mid M$ accepts x within $|y|$ steps $\}$. Clearly, D is decidable. Also, $x \in C$ if and only if there exists y such that $\langle x, y \rangle \in D$. Thus, $C = \{x \mid \exists y(\langle x, y \rangle \in D)\}$.

5. (Bonus Question) Show that the problem of determining whether a CFG generates all string in $\mathtt{1}^*$ is decidable. In other words, show that $\{\langle G \rangle \mid G$ is a CFG over $\{\mathtt{0}, \mathtt{1}\}$ and $\mathtt{1}^* \subseteq L(G)\}$ is a decidable language.

   Answer: Please discussed the solution with Yu-Han directly.