CS 682 (Spring 2001) - Solutions to Assignment 2

(1) Show that there exists a TM $M_i$ such that

\[ L(M_i) = \{M_i\}. \]

Proof. Let $f$ be recursive such that $M_{f(i)}$ is a machine that accepts $M_i$ alone. Such a machine is clearly computable in a uniform way. By the recursion theorem, there is some $i_0$ such that

\[ L(M_{i_0}) = L(M_{f(i_0)}) = \{M_{i_0}\}. \]

(2) Let $\mathcal{L}_i$ denote linearly-bounded automata. Show that

\[ \{\mathcal{L}_i | L(\mathcal{L}_i) \neq \Sigma^*\} \]

is an r.e. complete set.

Proof. Denote the set by $\Delta$. It is enough to construct a many-one reduction of a known r.e. complete set to $\Delta$, but we can easily construct a many-one reduction from any r.e. set $L(M)$ to $\Delta$ as follows:

Let $f$ be recursive such that $f(x)$ is the LBA $\mathcal{L}^x$ that rejects an input $y$ iff $y \in \text{VAL}(M)$ and the first configuration in $y$ has $x$ as the tape content.

\[ x \in L(M) \iff \text{there is such a } y \text{ in } \text{VAL}(M) \iff \text{there is a } y \text{ that } \mathcal{L}^x \text{ rejects} \iff L(\mathcal{L}^x) \neq \Sigma^* \iff \mathcal{L}^x \in \Delta. \]

(3) Show that

a) \( \{\mathcal{L}_i | L(\mathcal{L}_i) \text{ is cofinite}\} \equiv_m \{M_i | L(M_i) \text{ is finite}\} \)

b) \( \{\mathcal{L}_i | L(\mathcal{L}_i) \text{ is not regular}\} \equiv_m \{M_i | L(M_i) \text{ is infinite}\} \)

Proof.

a) For the left-to-right reduction, let $f$ be recursive such that $M_{f(i)}$ simulates $\mathcal{L}_i$ and accepts iff $\mathcal{L}_i$ rejects, so that $L(M_{f(i)}) = L(\mathcal{L}_i)$. Clearly

\[ L(\mathcal{L}_i) \text{ is cofinite} \iff L(M_{f(i)}) \text{ is finite}. \]

For the right-to-left reduction, let $f$ be recursive such that $\mathcal{L}_{f(i)}$ accepts $y$ iff $y$ is not a valid computation of $M_i$, so that $L(\mathcal{L}_{f(i)}) = \text{VAL}(M_i)$. Clearly

\[ L(M_i) \text{ is finite} \iff L(\mathcal{L}_{f(i)}) \text{ is cofinite}. \]
b) For the left-to-right reduction, let \( f \) be recursive such that \( M_{f(i)} \) accepts \( x \) iff for all \( j \leq x \), \( L(A_j) \neq L(L_i) \): for each \( j \leq x \), it simultaneously searches for a difference between the recursive sets \( L(A_j) \) and \( L(L_i) \); if such differences are found for all \( j \leq x \) it accepts \( x \), and if not it does not halt (no choice here). Then
\[
L(L_i) \text{ is not regular } \iff \forall j \ L(A_j) \neq L(L_i) \iff |L(M_{f(i)})| = \infty.
\]

For the right-to-left reduction, let \( f \) be recursive such that \( L_{f(i)} \) accepts \( \text{VAL}(M_i) \). Then
\[
|L(M_i)| = \infty \iff L(L_{f(i)}) \text{ is not regular}
\]

since for any \( M \), \( \text{VAL}(M) \) is regular iff it is finite (this is an easy consequence of the pumping lemma for regular languages). 

\[\blacksquare\]

(4) Let \( A = \{ M_i | |L(M_i)| = 2 \} \). Show that
a) \( A \leq_m \{ M_i | L(M_i) \text{ is finite} \} \)

b) \( A \leq_m \{ M_i | L(M_i) \text{ is infinite} \} \).

**Proof.**

a) Let \( f \) be recursive such that \( M_{f(i)} \) rejects \( x \) iff exactly 2 strings are enumerated by the \( x \)-th step of the recursive enumeration of \( L(M_i) \).

If \( |L(M_i)| = 2 \) then there is some \( x_0 \) such that both strings are enumerated after \( x_0 \) steps, and then \( M_{f(i)} \) will reject every \( x \geq x_0 \) and therefore \( L(M_{f(i)}) \) is finite.

If \( |L(M_i)| \neq 2 \) then there is some \( x_0 \) such that for all \( x > x_0 \) the number of strings enumerated in \( x \) steps is not 2 (if \( |L(M_i)| > 2 \) let \( x_0 \) be the number of steps it takes to enumerate 3 strings, and if \( |L(M_i)| < 2 \) any \( x_0 \) will do). \( M_{f(i)} \) will then accept all \( x > x_0 \), and therefore \( L(M_{f(i)}) \) is infinite.

b) Follow the same construction as in (a), but have \( M_{f(i)} \) accept instead of reject and vice versa. The same proof (with the expected alterations) applies.

\[\blacksquare\]