CS 682 (Spring 2001) - Solutions to Assignment 1

Throughout these solutions, input strings will be interchangeably interpreted as strings, numbers, or machine descriptions. Recall that every string over a finite alphabet can be regarded as a number, simply by choosing as the base the size of the alphabet. Also, the code for the i-th Turing machine $M_i$ can be assumed to be i itself, since the description of $M_i$ can be decoded from i simply by enumerating all Turing machine descriptions in the standard way and picking the i-th one. Therefore the set $\{M_{i_1}, M_{i_2}, \ldots \}$ can be regarded as the set $\{i_1, i_2, \ldots \}$, and $|M_i|$ can be assumed to be i.

(1) Prove that

\[ \{M_i \mid L(M_i) \text{ is not a regular set} \} \]

is a productive set.

**Proof.** Under the convention mentioned above, let $L_{\text{NREG}} = \{ i \mid L(M_i) \text{ is not a regular set} \}$. We have to define a recursive production function $p : \omega \to \omega$ with the property that $p(i) \in L_{\text{NREG}} \setminus L(M_i)$ for any i such that $L(M_i) \subseteq L_{\text{NREG}}$. That is, the function $p$ is constructive evidence that the language $L_{\text{NREG}}$ is not r.e. Notice that we do not place any restrictions on the values of $p(i)$ if $L(M_i) \not\subseteq L_{\text{NC}}$, although such values should be defined, since $p$ is required to be recursive, hence total.

The proof will use a slightly more elaborate diagonalization argument to ensure that the language $L(M_{p(i)})$ is not equal to any language accepted by a machine $M \in L(M_i)$. In addition, we will encode the halting problem into $L(M_{p(i)})$ so that the language is not regular. For the diagonalization argument to work, we need that $L(M_{p(i)})$ be infinite, which we cannot assume beforehand (and not decide either). So, we will “pad” $L(M_i)$ by infinitely many distinct machines accepting a fixed non regular set.

Consider the language $L_{01} = \{ 0^m1^m \mid m \geq 1 \}$, the standard example of a language that is not regular, but recursive. Let $L$ be an infinite r.e. set of machines accepting $L_{01}$. We can enumerate such a set $L$, for instance by taking one machine accepting $L_{01}$ and padding it with additional states.

Now, let i be given, and let $j_1, j_2, \ldots$ be an enumeration of $L(M_i) \cup L$ (since both $L(M_i)$ and $L$ are r.e., such an enumeration exists). First define a sequence $(x_k)$ as follows:

\[
\begin{align*}
x_1 &= \begin{cases} \text{the smallest } x \text{ such that } M_{j_1}(x) \downarrow & \text{if it exists} \\
\uparrow & \text{otherwise} \end{cases} \\
x_{k+1} &= \begin{cases} \text{the smallest } x > x_k + 1 \text{ such that } M_{j_k}(x) \downarrow & \text{if it exists and } x_k \downarrow \\
\uparrow & \text{otherwise} \end{cases}
\end{align*}
\]

Note that this sequence is computable. To compute $x_{k+1}$, we recursively compute $x_k$, and if the computation terminates, we start the search for the smallest x. If any part does not terminate, we are in the second case of the definition anyway. Now define $p$ to be a recursive function such that

\[
M_{p(i)}(x) = \begin{cases} 1 & \text{if there is a } k \text{ with } x = x_k + 1 \text{ and } M_k(k) \downarrow \\
\uparrow & \text{otherwise} \end{cases}
\]

We should verify that $p$ is recursive. We can compute the $x_k$, as described above, and when we find one with $x_k + 1 = x$, we run $M_k(k)$. If either we don’t find such a $k$, or $M_k(k) \uparrow$, $M_{p(i)}$ will not accept $x$. This is a machine involving quite a number of “sub-machines” (or sub-routines), and
so

By the recursion theorem, there is a fixed point automaton $A$ is recursive and $L$ is r.e.: to list the set, we try simultaneously to verify all of the inequalities

First note that since regular sets are recursive, the set $L$ is immune.

Proof. First, $ECO$ is minimal and $2$ is infinite r.e., then let $r(i)$ be an index of a machine which simulates the automaton $A_i$.

By the recursion theorem, there is a fixed point $i_0$ for $r$. But then

so $M_{i_0}$ and $A_j$ is a pair with the required properties.

(3) Define

$ECO_{\frac{1}{2}} = \{ M_i \mid 2|M_j| < |M_i| \implies L(M_i) \neq L(M_j) \}$.

Proof. First, $ECO_{\frac{1}{2}}$ is infinite because it contains the set of minimal Turing machines: if $M_i$ is minimal and $2j < i$ then certainly $j < i$ and consequently $L(M_i) \neq L(M_j)$, so $M_i \in ECO_{\frac{1}{2}}$. Suppose $L$ is an infinite r.e. subset of $ECO_{\frac{1}{2}}$. Define $f$ to be a recursive function such that $f(i) \in L$ and $f(i) > 2i$. Let $i_0$ be a fixed point for $f$. Then

contradicting the definition of $ECO_{\frac{1}{2}}$. Therefore $ECO_{\frac{1}{2}}$ does not contain any infinite r.e. sets, and is immune.