CSCI 510, Fall 2016, Homework # 4

SOLUTION SKETCHES

Due date: Wednesday, November 2, Midnight

1. Let A be a Turing-recognizable language consisting 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.

   Since A is Turing-recognizable, it is also Turing-enumerable, and so we assume that there is a machine to enumerate the machines M_i. Description of machine D:

   On input w:

   Let $\ell = |w|$

   Run the enumerator for A ℓ times to find machine M_ℓ.

   Run machine M_ℓ on w. (This must halt, since M_ℓ is a decider.)

   If M_ℓ accepts, *reject*, else *accept*.

   Machine D is a decider since it always halts. Machine D cannot be in the list that A enumerates, since for each machine $M_i \in A$ it gives different judgements on all strings of length i.

2. Let $A = \{\langle R\rangle |\ R$ is a regular expression describing a language containing at least one string w that has 111 as a substring (*i.e.* $w = x111y$ for some x and y $\}$. Show that A is decidable.

   We can turn any regular expression into a deterministic finite automaton. Now the problem has been reduced to finding whether there exists a path from the start state of the DFA to an accept state with 111 along the path somewhere.

   This can be solved easily with a graph search algorithm that starts at the root and expands paths until all reach accept states, or all start looping (if the length of the loop is less than 3, loop one or two more times). Examine all these paths to see if 111 shows up.

3. Let $A = \{\langle R, S\rangle |\ R$ and S are regular expressions and $L(R) \subseteq L(S)\}$. Show that A is decidable.

   Find DFAs for $L(R)$ and $L(S)$, and then find the DFA for the difference, $L(R) - L(S) = L(R) \cap \overline{L(S)}$. There are terminating algorithms for each of these steps. If this language is empty (which is decidable, see text), then $L(R) \subseteq L(S)$.