CS281 Spring 2010: Homework 2

Due Wednesday, April 21st – in class

1. **Exercise 9.1.3:** Here are two definitions of languages that are similar to the definition of L_d, yet different from that language. For each, show that the language is not accepted by a Turing machine, using a diagonalization-type argument. Note that you cannot develop an argument based on the diagonal itself, but must find another infinite sequence of points in the matrix suggested by Fig. 9.1. *Note from TA: if you do not have access to Hopcroft, e-mail me and I will send you a scan of the page in question.*

   a) The set of all w_i such that w_i is not accepted by M_{2i}.

   b) The set of all w_i such that w_{2i} is not accepted by M_i.

2. **Exercise 9.2.4:** Let $L_1, L_2, \ldots, L_k$ be a collection of languages over alphabet Σ such that:

   1. For all $i \neq j$, $L_i \cap L_j = \emptyset$; i.e. no string is in two of the lnaguages.
   2. $L_1 \cup L_2 \cup \ldots \cup L_k = \Sigma^*$; i.e., every string is in one of the languages.
   3. Each of the languages L_i, for $i = 1, 2, \ldots, k$ is recursively enumerable.

   Prove that each of the languages is therefore recursive.

3. **Exercise 9.2.6 (a) (b):** We have not discussed closure properties of the recursive langages or the RE languages, other than our discussion of complementation in Section 9.2.2. Tell whether the recursive languages and/or the RE languages are closed under the following operations. You may give informal, but clear, constructions to show closure.

   a) Union.

   b) Intersection.

4. **Exercise 9.3.7:** Show that the following problems are not recursively enumerable:

   a) The set of pairs (M, w) such that TM M, started with input w, does not halt.

   b) The set of pairs (M_1, M_2) such that $L(M_1) \cap L(M_2) = \emptyset$.

   c) The set of triples (M_1, M_2, M_3) such that $L(M_1) = L(M_2)L(M_3)$; i.e. the language of the first is the concatenation[1] of the languages of the other two TM's.

[1] Recall that the concatenation of two languages consists of the set of all concatenations of strings from those languages. Namely, $L_1 L_2 = \{xy | x \in L_1, y \in L_2\}$ and xy is understood to be just x if y is empty, just y if x is empty, and $a_1 a_2 \cdots a_r b_1 b_2 \cdots b_s$ when x and y are the strings composed of the symbols $a_1 a_2 \cdots a_r$ and $b_1 b_2 \cdots b_s$ respectively.