Theory of Computation

Homework 1

Due: 2013/10/01

Problem 1. Please describe the workings of the following two Turing machines:

a. Let M be the Turing machine $M = (K, \Sigma, \delta, s)$, where $K = \{s, h\}$

$p \in K$	$\sigma \in \Sigma$	$\delta(p, \sigma)$
s	$\triangleright$	$(s, \triangleright, \rightarrow)$
s	1	$(s, 0, \rightarrow)$
s	0	$(s, 1, \rightarrow)$
s	$\sqcup$	$(h, \sqcup, -)$

b. Let M be the Turing machine $M = (K, \Sigma, \delta, s_0)$, where $K = \{s_0, s_1, h\}$

$p \in K$	$\sigma \in \Sigma$	$\delta(p, \sigma)$
s_0	$\triangleright$	$(s_0, \triangleright, \rightarrow)$
s_0	1	$(s_1, 1, \rightarrow)$
s_0	0	$(s_0, 0, \rightarrow)$
s_1	0	$(s_0, 0, \rightarrow)$
s_1	1	$(h, 1, -)$
s_0	$\sqcup$	$(h, \sqcup, -)$
s_1	$\sqcup$	$(h, \sqcup, -)$

Solution:

a. This Turing machine converts all 1's in the binary input string to 0's and vice versa.

b. This Turing machine scans to the right until it finds two consecutive 1's and then halts; otherwise, it halts at the end of the binary input string.

□

Problem 2. Show that if a language is recursively enumerable, then there is a Turing machine that enumerates it (i.e., to output its members) without ever repeating an element of the language. Recall that in the original definition of enumeration on p. 41 of the slides, we do not require that every member is printed only once.

Solution: Suppose M is a Turing machine that enumerates (i.e., prints members of) L in some order. Construct M^* to enumerate elements in L with no repetitions as follows. M^* begins by simulating M and has a tape to record the printed elements. Whenever M is about to output a string, M^* will first check the tape to see if it has been outputted before. If the string is found on the tape, then M^* will not print it; otherwise, M^* will output the string and also record it on the tape, for future references. $\square$