6.045J/18.400J: Automata, Computability and Complexity Nancy Lynch

Homework 5

Due: March 12, 2007 *Elena Grigorescu*

Problem 1: **Detailed Turing machine description**
Give a complete, formal description of a (basic one-head, one-tape) Turing machine that *decides* the language $L = \{0^i 1^j : i > j\}$. This should consist of a list of the tuple components of the Turing machine, with the transition function represented by a state transition diagram. (Yes, we know that it's tedious to write Turing machine descriptions. But everyone should do it once. We'll try to avoid asking for this on other homeworks.)

Problem 2: **Higher level Turing machine description**
Describe the operation of a basic Turing machine that *recognizes* the language $L = \{ww^R w : w \in \{0,1\}^*\}$. This time, your description should not be completely formal. Rather, it should consist of a list of the machine tuple components, with the transitions described in careful English.

You may describe your construction in terms of a variant of the basic Turing machine model presented in class or in Sipser's book (e.g., multitape), provided that you quote the correct transformation result to show how one would turn your construction into a description of a basic Turing machine to recognize the same language.

Problem 3: **Turing-recognizability and Turing-decidability**

1. Sketch proofs that the class of Turing-recognizable languages is closed under the language operations union, intersection, concatenation, and star.
2. Sketch proofs that the class of Turing-decidable languages is closed under the language operations union, intersection, complement, concatenation, and star.

Problem 4: **Enumerability and decidability**
Suppose that L is a language over alphabet $\{0,1\}^*$. Prove the following:

1. If L is decidable then it can be enumerated by an enumerator Turing machine in strictly increasing order, with no repeats. (We order finite strings in order of length, and for the same length, in lexicographic order.)
2. If L can be enumerated in strictly increasing order, with no repeats, then L is decidable.

Problem 5: **Nondeterministic Turing machines**
Modify the proof idea on p. 150 of Sipser's book to prove that any language that is *decidable* by a nondeterministic Turing machine is also *decidable* in the usual sense (by a basic, deterministic Turing machine).
(You may assume the following fact about trees: If every node in a tree has only finitely many children and the tree has no infinite branches, then the entire tree has finitely many nodes.)