ECS120 Introduction to the Theory of Computation
Fall Quarter 2007

# Homework 7 Help
## Due Wednesday November 21, 2007

## Problem 7.1

**Explain why the following is not a description of a legitimate Turing machine.**

**$M_{\text{bad}}$ = "The input is a polynomial $p$ over variables $x_1, \ldots, x_k$.**

1. **Try all possible settings of $x_1, \ldots, x_k$ to integer values.**
2. **Evaluate $p$ on all of these settings.**
3. **If any of these settings evaluate to 0, *accept*; otherwise, *reject*."**

**Sipser Exercise 3.7**

No hints! Good luck :)

## Problem 7.2

**Show that the collection of Turing-recognizable languages is closed under the operation of:**

**(b) concatenation**

**(c) star**

**Sipser Problem 3.16 (b) and (c)**

Problem 3.16 (a) is solved in your book. It proves that Turing-recognizable languages are closed under union. This is similar to what we have seen before, except that now we are working with Turing machines.

## Problem 7.3

**Prove that a language $L$ is decidable if and only if some enumerator enumerates $L$ in lexicographic order.**

What is lexicographic ordering? A quick trip to the index in your book shows that the definition to this is on page 14.

Did you catch the "if and only if" in the question? This means you have two directions to prove:

1. If $L$ is decidable, then some enumerator enumerates $L$ in lexicographic order.
2. If some enumerator enumerates $L$ in lexicographic order, then $L$ is decidable.

How do you prove each direction? Start with what you are given. For example, lets consider the first half of the proof. We know $L$ is decidable. Definition 3.6 on page 142 of your book (which you should be reading!) states that:

> Call a language **decidable** if some Turing machine decides it.

Therefore, $L$ must have some Turing machine that decides it. Let $M$ be the Turing machine that decides $L$. Now, we must show that there exists an enumerator $M'$ which enumerates $L$ in lexicographic order. This part is up to you!

EXTRA HINT: For the second part, consider the cases of when $L$ is finite or infinite separately

## Problem 7.4

**Classify the following languages as decidable, acceptable, or neither, and give an algorithm (i.e. decision procedure) if they are decidable, or a TM if you think they are acceptable but not decidable.**

**(a)** $L_a = \{ \langle M \rangle : M$ **accepts some even-length string** $\}$

**(b)** $L_b = \{ \langle \alpha \rangle : \alpha$ **is the shortest regular expression for** $L(\alpha) \}$

**(c)** $L_c = \{ \langle M \rangle : M$ **is a C program that halts on** $\langle M \rangle \}$

**(d)** $L_d = \{ \langle M, w \rangle : M$ **is a TM which uses at most 11 tape squares when run on** $w \}$

CLEARLY MARK IF THE LANGUAGE IS DECIDABLE, ACCEPTABLE, OR NEITHER. Don't just provide a TM and let me "decide" if you are showing the statement is decidable or acceptable.

What should your answer look like? It should look similar to the theorems and proofs in section 4.1 of your book.