CSE396, Spring 2010 Problem Set 11 (the last) Due Mon. 4/26, in class

Schedule (as noted on PS10): The **Final Exam** is on **Tuesday, May 4**, **11:45am–2:45pm** in **Hochstetter 114**. It will be *open-book, open-notes*. It will cover all of the course material, Chapters 1–5 and 7, skipping the regular-languages Pumping Lemma and the Post Correspondence problem, but parts of Section 9.3 needed for Theorem 9.34 (the "alternative proof of the Cook-Levin theorem," 2nd. ed. only?). I intend to have an optional pre-exam "Review Session" on *Monday, May 3*, time and room still tba.

Reading: For Wednesday and the last Monday, read Chapter 7 and Section 9.3. Section 5.2 and all of Chapter 6 are skipped, while lecture will replace the long proof of the Cook-Levin Theorem in section 7.4 with the shorter proof in section 9.3. Note that much material in sections 7.1 and 7.2 has already been touched on, so lecture will focus on 7.3–7.4.

If you have a major assignment due April 24–26, you may e-mail me for an extension; I need to have these on record by Fri. 4/23, and the extension will in no case be later than Thu. 4/29, 11am.

(1) Show that $A_{TM} \leq_m K_{TM}$, where "K" is the standard name for the complement of the diagonal language "D." More precisely, $K_{TM} = \{$ TM's M: M does accept its own code M $\}$, and this is formally the complement of D_{TM} *in the "type" of valid TM's*. Hint—use the first reduction from the Monday 4/19 lecture, and explain why it accomplishes the logic of this case too. (12 pts.)

(2) Text, problem 5.33 on page 213: Show that the language $S = \{ M : L(M) = \{ M \} \}$ is neither c.e. not co-c.e. You can also think of S as the language of Java programs P that accept only their own code, and nothing else. (Note that end-of-file is an ASCII character, so even if P is distributed over multiple files and streams, what the Java compiler ultimately processes is a single string over the ASCII alphabet.) *Hint:* Show $K_{TM} \leq_m S$ by adapting the "all-or-nothing switch" that was used to reduce A_{TM} to K_{TM} in problem (1), and show $D_{TM} \leq_m S$ by adapting the switch in a different way. You may regard the instance "M" of the K_{TM} or D_{TM} problem as a "packet" inside the target machine M' of your reductions, and also as a "text string" to operate on... 18 pts. total)

(3) Show that the language $REGULAR_{CFG} = \{$ CFGs $G : L(G)$ is regular $\}$ is undecidable. Note that this is not the same as deciding whether the rules of G obey the format of a "regular grammar" as described in class. Rather, the question is whether the language of G is regular. Use the fact that the language of valid accepting computations of a single-tape TM is not regular, and do a reduction from E_{TM}. (18 pts.)

(4) Text, exercise 7.9 on p295 (1st ed. 7.10 on p272), i.e., show that the language of undirected graphs that have a triangle is decidable in polynomial time. Pseudocode with nested for-loops and an analysis of running time are expected—you may assume the graph G is encoded as an $n \times n$ adjacency matrix A_G where n is the number of nodes in G. You may state the running time in terms of "n," even though the actual size of the input is $N = n^2$. (12 pts.)

(5) Show that the problem, "Given a set of S of polynomial equations in variables $x_1, \ldots, x_n$ with integer coefficients, is there a solution to all equations with $x_1, \ldots, x_n$ all either zero or one?" is NP-complete. Show that the problem is in NP, and then use a polynomial-time mapping reduction from 3SAT. (*Hint:* convert each clause into an equation. 21 pts., for 81 points total on this last set.)