CS:5340 Homework 2

Due: Tue, 9/12

Notes: (a) Any problem numbers mentioned in the handout refer to problems in the textbook, by Arora and Barak. (b) It is possible that solutions to some of these problems are available to you via other theory of computation books or on-line lecture notes, etc. If you use any such sources, please acknowledge these in your homework *and* present your solutions in your own words. You will benefit most from the homework, if you seriously attempt each problem on your own first, before seeking other sources. (c) As mentioned in the syllabus, it is okay to form groups of two in solving and submitting homework solutions. But, my advice from (b) still applies: you will benefit most from the homework, if you seriously attempt each problem on your own first, before seeking help from your group partner. (d) Discussing these problems with any of your classmates is okay, provided you and your classmates are not being too specific about solutions. In any case, make sure that you take no written material away from these discussions *and* (as in (b)) you present your solutions in your own words. When discussing homework with classmates please be aware of guidelines on "Academic Dishonesty" as mentioned in the course syllabus.

1. An instance of the *Post Correspondence Problem* (PCP) is a collection P of "dominos:"

$$P = \left\{ \left[\frac{t_1}{b_1}\right], \left[\frac{t_2}{b_2}\right], \ldots, \left[\frac{t_k}{b_k}\right] \right\}.$$

   Each t_i and each b_i, $1 \leq i \leq k$ is some string from a finite alphabet Γ. The collection P is said to contain a *match* if there is a sequence $i_1, i_2, \ldots, i_\ell$ where

$$t_{i_1} t_{i_2} \cdots t_{i_\ell} = b_{i_1} b_{i_2} \cdots b_{i_\ell}.$$

   The problem is to determine if the given instance of PCP contains a match. One can think of PCP also as a function that maps instances of PCP that contain a match into 1 and the rest of the instances into 0. In 1946 Emil Post proved that PCP is not computable by any Turing machine. While you don't have to prove this here, below are a couple of problems that might help you appreciate PCP.

   (a) Find a match in the following instance of PCP:

$$\left\{ \left[\frac{ab}{abab}\right], \left[\frac{b}{a}\right], \left[\frac{aba}{b}\right], \left[\frac{aa}{a}\right] \right\}.$$

   (b) For the instance of PCP provided in part (a), you can think of the alphabet Γ as being $\{a, b\}$. Show that if the alphabet Γ is restricted to be $\{1\}$, then PCP is computable. You do not have to describe a Turing machine for this proof; a clearly stated algorithm in pseudocode with comments will suffice.

2. We say that a Turing machine M *accepts* a string $w \in \{0,1\}^*$ if on input w, M halts and outputs 1. A Turing machine M is said to have property R if whenever M accepts w it accepts w^R. (**Note:** w^R denotes the string obtained by reversing string w; e.g., $(\texttt{011})^R$ is $\texttt{110}$.) Define a function $R : \{0,1\}^* \rightarrow \{0,1\}$ as follows: $R(\alpha) = 1$ if M_α has property R and $R(\alpha) = 0$ otherwise. Prove that the function R is uncomputable.

3. Problem 1.12 (Chapter 1, Page 35).
4. Problem 1.15(b) (Chapter 1, Page 37).
