Computability and Complexity — Winter 2009

Problem Set #5 — Instructor: Anna Karlin

Due on **February 20, 2009** by email to `ncthach@cs.washington.edu`

Instructions: You are allowed to discuss the problems with fellow students taking the class. However, you must write up your solutions completely on your own. Moreover, if you do discuss the problems with someone else, I am asking, on your honor, that you do not take any written material away from the discussion. In addition, for each problem on the homework, I ask that you acknowledge the people you discussed that problem with, if any.

The problems have been carefully chosen for their pedagogical value and hence might be similar or identical to those given out in past offerings of this course at UW, or similar courses at other schools. Using any pre-existing solutions from these sources, from the Web or other textbooks constitutes a violation of the academic integrity expected of you and is strictly prohibited.

Most of the problems require only one or two key ideas for their solution – spelling out these ideas should give you most of the credit for the problem even if you err in some finer details. So, make sure you clearly write down the main idea(s) behind your solution.

A final piece of advice: Begin work on the problem set early and don't wait till the deadline is a day or two away.

Readings: Arora/Barak Chapters 7, 8, Sipser Sections 8.1 – 8.3, 1 10.2 and 10.4

1. Show that, if 3-SAT is PSPACE-hard, then PSPACE=NP.

2. Consider the problem *In-Place-Acceptance*: Given a single tape deterministic Turing machine M and an input x, does M accept x without the head ever leaving the first $|x|+1$ cells of its string?

   Prove that *In-Place-Acceptance* is PSPACE-complete.

   Hint: Do a direct reduction (from an arbitrary problem in PSPACE). Consider padding the input.

3. For fun probability problem: There is an event in an n-seat auditorium and n shy people have each been assigned one of the n seats. Unfortunately, the first person to walk in is the Absent-Minded Professor, who chooses a seat at random and sits there. People walk in one by one, and each person who walks in after the professor sits down in his or her own seat if it is open. However, if someone is already sitting there, s/he is too shy to protest, and picks an empty seat at random and sits there. What is the probability the last person will sit in his/her assigned seat? What is the expected number of people that will end up in their assigned seat?

4. Consider the following distributed computing scenario:

   A *process* is a directed graph $G = (V, E)$ whose vertices are called *states* and whose edges are called *transitions*. A *system of communicating processes* is a set of processes $\{G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots, G_n = (V_n, E_n)\}$, where all the V_i's are assumed to be disjoint, together with a set $P = \{(e_1, e'_1), \ldots, (e_m, e'_m)\}$ of unordered pairs of transitions called *communication pairs.* Each communication pair $(e_i, e'_i) \in P$ is such that $e_i \in E_j$ and $e'_i \in E_k$ for some $j \neq k$. Intuitively, a pair of transitions in P is one way whereby the corresponding processes can communicate, by synchronously changing their state according to the two transitions. For such communication to take place, the two processes must be at the appropriate states (the tails of the two transitions).

The *set of system states* V of a system of communicating processes is the Cartesian product of all V_i's: $V = V_1 \times V_2 \times \ldots \times V_n$. We define the *transition relation* $T \subseteq V \times V$ as follows: $((a_1, a_2, \ldots, a_n), (b_1, b_2, \ldots, b_n)) \in T$ if and only if there are two indices $j \neq k$ such that $\{(a_j, b_j), (a_k, b_k)\} \in P$ and $a_i = b_i$ for all i other than j and k. That is, $((a_1, a_2, \ldots, a_n), (b_1, b_2, \ldots, b_n)) \in T$ if and only if we can go from $(a_1, a_2, \ldots, a_n)$ to $(b_1, b_2, \ldots, b_n)$ by changing two components of the system state according to a pair in P, and keeping all other components the same.

Designing systems of communicating processes so that they conform to various specifications and testing that the result of the design is correct are important problems. Unfortunately, just about all important properties of such systems are intractable to test. For example, let us define a *deadlock system state* to be a system state $d \in V$ such that there is no $a \in V$ with $(d, a) \in T$. Of particular interest are deadlock states that are actually reachable. (Only a small fraction of the system states can be accessed in actual operation; unreachable deadlock states are therefore irrelevant). This leads us to the following problem.

Reachable Deadlock: Given a system of communicating processes, and an initial system state a, is there a deadlock system state d that is reachable from a in the transition relation?

Prove that *Reachable Deadlock* is in PSPACE.

5. *Extra Credit:* Prove that *Reachable Deadlock* is PSPACE-hard. (Combined with the result of the previous problem, this means it is PSPACE-complete.)