Computability and Complexity — Winter 2009

Problem Set #2 — Instructor: Anna Karlin

Due on **January 30, 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 1 and 2, Sipser Chapters 3-5.

1. Recall that a Turing machine enumerator M is a two-tape Turing machine where both tapes are initially empty. The first tape is a work tape and the second is a write-only output tape. The enumerator M runs forever and in the process outputs a string $w_1\sharp w_2\sharp\ldots$ on its output tape where each $w_i \in \Sigma^*$ and $\sharp \notin \Sigma$. The set $\{w_1, w_2, w_3, \ldots\}$ is the language enumerated by the machine. In class, we argued (and it is also in Sipser - Theorem 3.21) that a language is enumerated by a Turing machine enumerator if and only if it is Turing recognizable.

   - Suppose a Turing machine enumerator outputs $w_1\sharp w_2\sharp\ldots$ with the property that for all i, $|w_i| < |w_{i+1}|$. Argue that the language enumerated by M is decidable.
   - Use the result you just showed and the equivalence of Turing enumeration and Turing recognition to show that any infinite Turing recognizable langauge has an infinite decidable subset.

2. Which of the following problems are decidable? Justify each answer:

   - Given Turing machines M and N, is $L(N)$ the complement of $L(M)$?
   - Given a Turing machine M, integers a and b, and input x, does M run for more than $a|x|^2 + b$ steps on input x?
   - Given a Turing machine M, does M have the property that M accepts w^R whenever it accepts w?

3. As observed in class, if our definition of a Turing machine (or programs in your favorite language) restricted the tape size (or equivalently the memory) to be finite, then we could

solve the halting problem. More precisely, suppose we define our Turing machines so that they have a single read/write tape of size at most 2^{34} cells. (This is approximately the amount of RAM a typical desktop computer might have now – 2 GB). Under this restriction on Turing machines, the halting problem is decidable. Give an estimate T of the number of steps it would take to decide the Halting problem (i.e., does $P(P)$ halt?) for Turing machines P with at most 2^{10} states and alphabet equal to 0,1 and blank. Roughly, how long would it take for a computer executing a billion steps per second to execute T steps?

4. *Extra Credit*

   Give a proof that $\{0^n 1^n | n \geq 1\}$ is not regular. (Hint: There is a proof similar to the one we did for the palindromes.)

5. *Extra Credit*

   - Prove that a language A is Turing recognizable if and only if A is mapping reducible to A_{TM}.
   - Prove that a language B is decidable if and only if B is mapping reducible to

     $$\{0^n 1^n | n \geq 1\}.$$