Problem Set 8 — Due Tuesday, February 28, 2006

Problem 1 A k-pebble machine is a Turing machine that has two one-way infinite tapes — an input tape and an auxiliary tape. The machine cannot write on either tape, but it can place k (identical) pebbles on the auxiliary tape. It can place a pebble on the tape square it is scanning, it can pick up a pebble from the square it is scanning, and it can tell whether or not the square it is scanning contains any pebbles. (You can imagine that, initially, all of the pebbles are on the leftmost tape square, but as the machine computes it can push those pebbles around.) Assume that you can sense the presence of the left edge of the tape.

Show that, for some constant k, the following is true: if L is Turing acceptable, then there is a k-pebble machine that accepts L.

Try to use as few pebbles as possible.

Problem 2. For each language, indicate if it is:

decidable:	the language is Turing-decidable;
acceptable:	the language is Turing-acceptable, but you can't see that the language is Turing-decidable; or
not acceptable:	you can't see that the language is Turing-acceptable.

If you indicate that a language is decidable, sketch the algorithm. If you indicate that a language is acceptable, sketch the TM that shows this.

1. $L_1 = \{\langle M \rangle : M$ has a prime number of states$\}$
2. $L_2 = \{\langle M \rangle : M$ halts on some string$\}$
3. $L_3 = \{\langle M \rangle : M$ loops on some string$\}$
4. $L_4 = \{\langle M \rangle : M$ loops on every string$\}$
5. $L_5 = \{\langle M \rangle : M$ can enter state q_{27} (for some input)$\}$
6. $L_6 = \{\langle M \rangle : M$ never enters state q_{27} (regardless of initial input)$\}$
7. $L_7 = \{\langle M_1, M_2 \rangle : L(M_1) = L(M_2)\}$
8. $L_8 = \{\langle M_1, M_2 \rangle : L(M_1)$ is Turing-decidable$\}$
9. $L_9 = \{\langle M_1 \rangle : L(M_1)$ is Turing-acceptable$\}$
10. $L_{10} = \{\langle M \rangle :$ there is a regular language R such that $R \subseteq L(M)\}$

Problem 3. Consider a *black-box-L Turing machine* M^L. Such a machine is just like an ordinary Turing machine, except that it has an "oracle" for the language L. That means that the Turing machine has a special *query tape* and, if the TM writes on that tape a string x and goes into a special *query state*, q_{query}, then the next state will be either q_{yes} or q_{no}, depending on whether or not x is in L.

Part A. Show that $M^{A_{\text{TM}}}$ can decide any acceptable language.

Part B. Prove that there is a natural language L that the black-box Turing machine $M^{A_{\text{TM}}}$ can *not* decide.

Problem 4. As you may know, the CS and CSE majors have had a recent and dramatic decline in enrollments; following a national trend, our enrollments for CS+CSE fell from 949 in December of 2001 to 505 in December of 2005. And enrollments are still declining, it appears. Naturally, this has people concerned.

On behalf of the department, we would like to ask you: what might the CS Department do to make our undergraduate program more attractive, and to retain our students (especially our best students)? Please write up your ideas in a 1 page essay (more or less). Your essays will be shared with the Department's Undergraduate Study Committee, and selected write-ups will be given to the Chair and/or the the Department at large. This is a good opportunity to have your voice heard and to help out the Department.

Please typeset your essay. It should be written on a separate page that can can be detached from the rest of your homework solutions. That page may be anonymous, if you wish. You're encouraged to discuss this problem with others, and you may turn in essays having one or two authors. Please be frank (don't worry someone won't "like" what you have to say). Try to make your suggestions as useful as possible.

On behalf of the department, thanks in advance for your input.