6.080 / 6.089 Great Ideas in Theoretical Computer Science
Spring 2008

6.080/6.089 Problem Set 2

Assigned: Thursday, Feb. 28, 2008 / Due: Thursday, March 13, 2008

1. In 1962, Tibor Rado defined $S(n)$, or the n^{th} "Busy Beaver shift number," to be the maximum number of steps made by any n-state Turing machine that eventually halts. Here a Turing machine has a two-way infinite tape with either 0 or 1 on each square, and all tape squares are initially set to 0. A "step" consists of writing a 0 or 1 to the current square, moving either left or right by one square, and either transitioning to a new state or halting (with all of these decisions determined by the current state together with the symbol on the current square).
   (a) Show that $S(1) = 1$.
   (b) Show that $S(2) \geq 6$. [*Hint:* Try various 2-state Turing machines until you find one that runs for 6 steps before halting.]
   (c) Show that $S(n)$ grows faster than any computable function. In other words, there is no computable function C such that $C(n) \geq S(n)$ for all n.
   (d) Show that there is not even a computable function C such that $C(n) \geq S(n)$ for infinitely many n.

2. Given a set of strings $L \subseteq \{0,1\}^*$, we say L is *computable* if there exists a Turing machine that, given as input a string x, decides whether $x \in L$. We say L is *c.e.* (for "computably enumerable") if there exists a Turing machine M that, when started on a blank tape, lists all and only the strings in L. (Of course, if L is infinite, then M will take an infinite amount of time.)
   (a) Let $HALT$ be the set of all Turing machines that halt when started on a blank tape. (Here each Turing machine is encoded as a binary string in some reasonable way.) Show that $HALT$ is c.e. [*Note:* In this and the following problems, you do not need to construct any Turing machines; just give a convincing argument.]
   (b) Let L be any c.e. set. Show that L is computable given an oracle that, for any string x, decides whether $x \in HALT$.
   (c) Show that a set L is computable if and only if L and $\overline{L}$ are both c.e. (Here $\overline{L}$ is the *complement* of L: that is, the set of all $x \in \{0,1\}^*$ such that $x \notin L$.)

3. Given a formal system F, recall that $\operatorname{Con}(F)$ is a mathematical encoding of the claim that F is consistent: in other words, that F never proves both that a statement is true and that it's false. Consider the "self-hating system" $F + \neg\operatorname{Con}(F)$: that is, F plus the assertion of its own inconsistency. Show that if F is consistent, then $F + \neg\operatorname{Con}(F)$ is an example of a formal system that is consistent but not sound. [*Note:* You can assume the Incompleteness Theorem.]

4. Let a *XOR-circuit of size* n be a circuit built entirely out of two-input XOR gates, which maps n input bits to n output bits. Also, call two circuits *equivalent* if they produce the same output whenever they're given the same input.
   (a) Show that, for every XOR-circuit of size n, there's an equivalent XOR-circuit with at most $n(n-1)$ gates.
   (b) Show that for every n, there's some XOR-circuit of size n such that every equivalent XOR-circuit has $\Omega\left(n^2/\log n\right)$ gates.

5. Suppose a Turing machine M has s internal states, and visits at most n different tape squares. Prove an upper bound (in terms of n and s) on the number of steps until M halts (assuming it does halt).