Problem Set 7 – Due Friday, May 17, 2013

For this problem set, please work in teams of 2–3 people. Submit one solution per team.

Problem 1. Design a Turing machine that takes as input a string of the form \( x \# y \), where \( x, y \in \{0, 1\}^+ \), and replaces the tape contents by the sum of binary numbers \( x \) and \( y \), again written as a binary number. Rather than following the conventions of your book, please employ those of the website [http://morphett.info/turing/turing.html](http://morphett.info/turing/turing.html). In particular, assume a two-way infinite tape. Try to make your program use as few rules as possible, measured by the number of 5-tuples that you need. Test your machine on plenty of inputs.

I don’t care what your machine does if presented an ill-formed string. I don’t care if your machine can produce strings with leading zeros, which might happen if \( x \) or \( y \) have leading zeros.

For grading this problem, please mail your solution, in the runnable format of the website above, to tcprovam@ucdavis.edu. A comment at the top of your program should list the names of the team members, in alphabetical order by last name, and the number of rules you used. A student solution will be used for our problem-set solutions.

A prize will go to the (apparently correct) solution with the fewest number of rules.

Problem 2. A TM \( M = (Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R) \) is oblivious if, when we run \( M \) on an input \( x \in \{0, 1\}^n \), the position of the head at step \( t \) depends only on \( t \) and \( n \). This must hold for all time steps \( t \) until \( M \) halts. Prove the following: for any TM \( M \) there exists an oblivious TM \( M' \) that decides the same language.

Problem 3. Classify each of the following languages as either (a) decidable—I see how to decide this language; (b) r.e.—I don’t see how to decide this language, but I can see a procedure to accept it; (c) co-r.e.—I don’t see how to decide this language, but I can see a procedure to accept its complement; or (d) neither: I don’t see how to accept this language or its complement. No justification is needed for your answers.

Part A. \( \{\langle M \rangle : M \text{ is a TM that accepts some string of prime length}\} \).
Part B. \( \{\langle M \rangle : M \text{ is a TM and } M \text{ has 100 states}\} \).
Part C. \( \{\langle M \rangle : M \text{ is a TM and } L(M) \text{ is regular}\} \).
Part D. \( \{\langle M \rangle : M \text{ is a TM and } L(M) \text{ is r.e.}\} \).
Part E. \( \{x : x \text{ is a C-program (no I/O or library calls) that halts on } x\} \).
Part F. \( \{\langle M \rangle : M \text{ is a TM and } M \text{ will visit state } q_{20} \text{ when run on some input } x\} \).
Part G. \( \{\langle G \rangle : G \text{ is a CFG and } G \text{ accepts an odd-length string}\} \).