

# Homework problems #32–37

CSC 341

due May 2, 2007

## Problem 32

A set  $S$  of natural numbers is *recursively reducible* to a set  $T$  of natural numbers if, and only if, there is a singularly, recursive function  $f$  such that, for every natural number  $n$ ,  $n \in S \Leftrightarrow f(n) \in T$ . Prove that if  $T$  is recursive and  $S$  is recursively reducible to  $T$ , then  $S$  is recursive.

## Problem 33 (for extra credit)

This problem is the same as the extra-credit problem 27, with the additional constraint that your solution should use Theorem 6.3. As in problem 27, we consider only Turing machines with the input alphabet  $\{1\}$  and the tape alphabet  $\{1, \sqcup\}$ .

The *run* of a Turing machine  $M$ , started on an input string  $s$ , is the number of transitions that  $M$  makes before halting. (If  $M$  does not halt at all on input  $s$ , then we'll stipulate that its "run" is defined to be 0.)

Let **longest-run-on-empty** be a singularly function that, given a number  $n$ , yields the largest run of any  $(n + 2)$ -state Turing machine on the input string  $\varepsilon$ . (The extra two states are the accept and reject states, so in effect  $n$  counts the number of states that the machine can be in before it halts.)

Use the recursion theorem to prove that no Turing machine computes **longest-run-on-empty**, in the sense that, when started on input  $1^n$ , it eventually halts with a string of **longest-run-on-empty**( $n$ ) 1s at the left end of its (otherwise blank) tape and the read-write head on the leftmost cell of the tape.

## Problem 34

Let  $t_1, t_2 : \Sigma^* \rightarrow \Sigma^*$  be computable functions. Show that there are Turing machines  $F_1$  and  $F_2$  such that  $t_1(\langle F_1 \rangle)$  is the encoding for a Turing machine equivalent to  $F_2$  and  $t_2(\langle F_2 \rangle)$  is the encoding for a Turing machine equivalent to  $F_1$ . (For this problem, a string that does not correctly encode any Turing machine should be understood as denoting a Turing machine that rejects every input, as in Theorem 6.8.)

## Problem 35

For any string  $x$ , let  $x^{\mathcal{D}}$  be the string formed by replacing each symbol in  $x$  with two copies of itself. (For instance, if  $x$  is 01001, then  $x^{\mathcal{D}}$  is 0011000011.) Prove that there is some natural number  $c$  such that, for every string  $x$  on the alphabet  $\{0, 1\}$ ,  $K(x^{\mathcal{D}}) \leq K(x) + c$ .

## Problem 36

A string  $s$  on the alphabet  $\{0, 1\}$  is “nearly balanced” if, and only if, the number of 0s in  $s$  differs from the number of 1s in  $s$  by no more than 8 (in either direction). Prove that there is some positive integer  $n_0$  such that all nearly balanced strings of length  $n_0$  or more are 1-compressible.

## Problem 37

Prove that every finite language is decidable in polynomial time.

Copyright © 2007 John David Stone

This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 United States License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/us/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.