

# Homework problems #17–22

CSC 341

due March 7, 2007

## Problem 17

Natural numbers  $i$  and  $j$  are said to be *equivalent modulo  $n$*  if, and only if, they differ by a multiple of  $n$ . In the notation we have developed for recursive function theory, define a predicate `equivalent-modulo?` that takes  $i$ ,  $j$ , and  $n$  as arguments and returns 1 if  $i$  and  $j$  are equivalent modulo  $n$  and 0 otherwise.

## Problem 18

Prove that the singular predicate `prime?`, which returns 1 if its argument is a prime number and 0 otherwise, is primitive recursive. (For purposes of this exercise, neither 0 nor 1 is prime.)

## Problem 19

The predicate `member?` takes two arguments, of which the first is the encoding for a list, and returns 1 if the second argument is equal to any of the elements in the list and 0 if it is not. Prove that `member?` is primitive recursive.

## Problem 20

The function `string-reverse` takes two arguments, of which the first is the size of an alphabet and the second is the encoding for a string in that alphabet, and returns the encoding for a string on the same alphabet that

is the reverse of the given string. Prove that `string-reverse` is primitive recursive.

## Problem 21

An "infinite bitstream" is a non-terminating stream in which every element is either 0 or 1. Prove by diagonalization that there are more infinite bitstreams than natural numbers.

## Problem 22 (for extra credit)

Design, write, and test a Scheme module containing a structured data type for partial recursive functions, analogous to the data-type modules that I wrote for deterministic and nondeterministic finite automata and for context-free grammars. For further extra credit, write an evaluator—a procedure that takes a partial recursive function structure as its first argument and additional arguments equal in number to the valence of the function represented by that structure, and returns the value of the function it describes. (Test the evaluator carefully, recalling that some partial recursive functions do not yield values for some arguments.)

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.