Fall 2021 CIS 511

Introduction to the Theory of Computation
Jean Gallier

Homework 5

November 10 2021; Due November 24, 2021

Problem B1 (40 pts). (1) Prove that the function, $f\colon \Sigma^* \to \Sigma^*$, given by

$$f(w) = w^R$$

is RAM computable. ($\Sigma = \{a_1, \ldots, a_N\}$).

(2) Prove that the function, $f\colon \Sigma^* \to \Sigma^*$, given by

$$f(w) = www$$

is RAM computable. ($\Sigma = \{a_1, \ldots, a_N\}$).

For simplicity, you may assume that $N = 2$.

You *must* run your interpreter from B2 on these two RAM programs for a few inputs. Show the two RAM programs as specified in the syntax of your interpreter in B2.

Problem B2 (80 pts). Write a computer program implementing a RAM program interpreter. You may want to assume that the instructions have five fields

N	X	`opcode`	j	Y
N	X	`opcode`	j	$N1$

with $j \in \{1, \ldots, k\}$, where k is the number of symbols in Σ, and that the opcodes are

`add` `tail` `clr` `assign` `gotoa` `gotob` `jmpa` `jmpb` `continue`

where `gota` corresponds to jump above, `gotob` to jump below, `jpma` corresponds to jump above if condition is satisfied, and `jmpb` to to jump below if condition is satisfied. Depending on the opcode, some of the fields may be irrelevant (set them to 0).

The number of input registers is n (so your memory must have at least n registers), and the total number of registers is p. The number k, n p are input to your interpreter, as well as the program to be executed (a sequence of instructions). Assume that line numbers are integers. Also, to simplify matters, you may assume that you only consider alphabets of size at most 10, so that $a_1, \ldots, a_k$ ($k \leq 10$) are represented by the digits $0, 1, \ldots, 9$.

Your program should output.

1. The input RAM program P
2. The input strings $w_1, \ldots, w_n$ to the RAM program P.
3. The value of the function being computed.
4. The sequence consisting of the memory contents and the current program counter as your interpreter executes the RAM program.

Test your interpreter on several RAM programs (and input strings), including the programs of B1.

To give you an idea of an implementation of this interpreter in `Matlab` here is the beginning of my program.

```
%
% RAM interpreter
%
% opcodes are coded numerically as follows:
%
% add = 1; tail = 2; clr = 3; assign = 4; gotoa = 5; gotob = 6; jmpa = 7;
% jpmb = 8; continue = 9
%
% Instructions have 5 fields
%  N      X    opcode    j    Y
%  N      X    opcode    j    N1
% where j corresponds to symbol a_j
% There are n input registers, a total number of p registers, and the
% alphabet size is k; symbols are coded as 1, 2, ..., k
% line numbers are nonnegative; unused line numbers are negative
% The registers are numbers 1, 2, ..., p
% The input RAM progrm is in RAMprog
% The program counter is pc
% input is a list indata  containing the n input strings
%
%
function [res, pc, regs, counter] = RAMinterp(RAMprog, n, p, k, indata)
lenprog = size(RAMprog, 1);
%
% insert your code here
%
```

To run this program I used the following input file.

```
%
```

```
%  Running RAM interpreter
%

% concatenation of two strings

indata1{1} = [1     2     1     2]
indata1{2} = [2     1     2     1     1     2     2]

[res, pc, regs] = RAMinterp(RAMconcat,2,4,2,indata1)

% string reversal

indata2{1} = [1     2     2     2     2     1     1     2     1     2]
indata3{1} = [2     2     2     2     2     1     1     1     1]

[res, pc, regs] = RAMinterp(RAMrev,1,4,2,indata2)

% f(w) = www

[res, pc, regs, counter] = RAMinterp(RAMtriple,1,5,2,indata3)
```

Here is the program to concatenate two strings.

```
%
%  a RAM program to concatenate two strings
%

RAMconcat =
[-1    3     4    0   1;
 -1    4     4    0   2;
 0     4     8    1   1;
 -1    4     8    2   2;
 -1    0     6    0   3;
 1     0     1    1   3;
 -1    0     2    0   4;
 -1    0     5    0   0;
 2     0     1    2   3;
 -1    0     2    0   4;
 -1    0     5    0   0;
 3     1     4    0   3;
 -1    0     9    0   0]
```

Problem B3 (30 pts). (1) Prove that the extended pairing function $\langle x_1, \ldots, x_n \rangle_n$ defined in the notes (see Definition 3.2 of the notes) satisfies the equation

$$\langle x_1, \ldots, x_n, x_{n+1} \rangle_{n+1} = \langle x_1, \langle x_2, \ldots, x_{n+1} \rangle_n \rangle.$$

Compute $\langle 2, 5, 7, 17 \rangle_4$ (this integer has 10 digits).

(2) Prove that $\langle x, 0 \rangle = \langle x, 0, \ldots, 0 \rangle_n$ for all $n \geq 2$ and all $x \in \mathbb{N}$.

(3) Prove that

$$\langle \Pi(1, n, z), \ldots, \Pi(n, n, z) \rangle_n = z$$

for all $n \geq 1$ and all $z \in \mathbb{N}$.

Problem B4 (30 pts). *Ackermann's function* A is defined recursively as follows:

$$\begin{aligned} A(0,\, y) &= y + 1, \\ A(x+1,\, 0) &= A(x,\, 1), \\ A(x+1,\, y+1) &= A(x,\, A(x+1, y)). \end{aligned}$$

Prove that

$$\begin{aligned} A(0,\, x) &= x + 1, \\ A(1,\, x) &= x + 2, \\ A(2,\, x) &= 2x + 3, \\ A(3,\, x) &= 2^{x+3} - 3, \end{aligned}$$

and

$$A(4, x) = 2^{2^{\cdot^{\cdot^{\cdot^{2^{16}}}}}} \Big\}^{x} - 3,$$

with $A(4, 0) = 16 - 3 = 13$. Equivalently (and perhaps less confusing)

$$A(4, x) = 2^{2^{\cdot^{\cdot^{\cdot^{2^{2}}}}}} \Big\}^{x+3} - 3.$$

Problem B5 (10 pts). Let $f \colon \mathbb{N} \to \mathbb{N}$ be a total computable function. Prove that if f is a bijection, then its inverse f^{-1} is also (total) computable.

TOTAL: 190 points.