CECS 329, Learning Outcome Assessment 9, April 13th, Spring 2023, Dr. Ebert

NO NOTES, BOOKS, ELECTRONIC DEVICES, OR INTERPERSONAL COMMUNICATION ALLOWED. Submit solutions to at most 2 LO problems on separate sheets of paper.

Problems

LO5. Let x be a natural number, and let reverse(x) denote the number whose binary representation is that of x's written backwards. For example, $(12)_2 = 1100$ and so reverse$(12) = 3$, since

$$(1100)^r = 0011 = (2+1)_2 = (3)_2.$$

As another example, $(26)_2 = 11010$ and so reverse$(x) = 11$, since $(11)_2 = 01011$, Show that reverse(x) is primitive recursive. You may use any PR functions from the General Models of Computation lecture examples and exercises, including the function bit(x, i) which equals the i th bit of x when x is written as a binary number, and len(x) which gives the length of x when written as a binary number. Hint: you do *not* have to provide a recursive definition.

Solution.

$$\text{reverse}(x) = \sum_{i=0}^{\text{len}(x)-1} \text{bit}(x, \text{len}(x) - i - 1)2^{\text{bit}(x,\text{len}(x)-i-1)}.$$

LO6. Do the following.

(a) Compute the Gödel number for program $P = J(1, 2, 3), S(5), Z(3), T(5, 2)$. Write your answer as a sum of powers of two minus 1.

Solution.

$$\beta(J(1, 2, 3)) = 4\xi(0, 1, 2) + 3 = 4\pi(\pi(0, 1), 2) + 3 = 4\pi(2, 2) + 3 = 4(19) + 3 = 79.$$

$$\beta(S(5)) = 4(4) + 1 = 17.$$

$$\beta(Z(3)) = 4(2) = 8.$$

$$\beta(T(5, 2)) = 4\pi(4, 1) + 2 = 4(47) + 2 = 190.$$

$$\gamma(P) = \tau(79, 17, 8, 190) = 2^{79} + 2^{97} + 2^{106} + 2^{297} - 1.$$

(b) Provide the URM program P whose Gödel number equals

$$2^{39} + 2^{230} + 2^{240} + 2^{269} - 1.$$

Solution. $\gamma^{-1}(x) = J(2,1,3), T(5,2), S(3), Z(8)$.

LO7. Answer and solve the following.

(a) When simulating a program P_x on input y via a universal program P_U, it is essential that P_U have access to each of the following values during the entire simulation of $P_x(y)$, *except* for

i. **X**input y.

ii. the number of instructions s of P_x.

iii. the index pci for where to locate the program counter in a configuration of $P_x(y)$.

iv. All of the above are essential values that P_U must access during the entire computation.

(b) A universal program P_U is simulating a program that has 137 instructions and whose Gödel number is

$$x = 2^8 + 2^{22} + 2^{726} + 2^{751} + 2^{c_5} + \cdots + 2^{c_{137}} - 1.$$

If the current configuration of the computation of P_x on some input has encoding

$$\sigma = 2^5 + 2^7 + 2^{13} + 2^{16} + 2^{20} - 1,$$

then provide the next configuration of the computation *and* its τ encoding. Hint: $176 = 16 \times 11$.

Solution.

Current configuration: $\tau^{-1}(\sigma) = (5,1,5,2,3)$, Program counter equals 3, and $\beta(I_3) = 726 - 22 - 1 = 703$ which implies a jump instruction, since $703 \bmod 4 = 3$. Moreover, $700/4 = 175$.

$$175 + 1 = 176 = 16 \times 11 = 2^4(2(5) + 1).$$

Moreover,

$$4 + 1 = 5 = 2^0(2(2) + 1)$$

and so $I_3 = J(1,3,6)$. Finally, since $R_1 = R_3$, the next configuration is

$$(5,1,5,2,6),$$

and

$$\sigma_{\text{next}} = \tau(5,1,5,2,6) = 2^5 + 2^7 + 2^{13} + 2^{16} + 2^{23} - 1.$$

LO8. Do the following.

(a) In one or more complete sentences, describe what is asserted by the Church-Turing Thesis.

Solution. Any [functional] deterministic step-by-step procedure that can be executed by a human using pencil and paper can be represented by a URM program.

(b) Consider the function

$$g(x) = \begin{cases} 1 & \text{if } \phi_x \text{ is total} \\ 0 & \text{otherwise} \end{cases}$$

In other words, $g(x) = 1$ iff program P_x halts on all of its inputs. We want to prove that $g(x)$ is undecidable, meaning there is no URM program that computes g. To do this, let's assume that $g(x)$ is computable by some URM program G. Then define the function $f(x)$ as follows.

$$f(x) = \begin{cases} P_x(x) + 1 & \text{if } g(x) = 1 \\ 0 & \text{if } g(x) = 0 \end{cases}$$

Use part a and an informal description of how you would go about computing $f(x)$ to establish that $f(x)$ is URM computable.

Solution. Simulate program G on input x to obtain $g(x)$. If the output is 0, then return 0. If the output is 1, then simulate program P_x on input x and return $P_x(x) + 1$.

(c) Let e denote the Gödel number of the URM program F that computes $f(x)$ from part b. In other words $F = P_e$. Show that a contradiction arises when we try to compute $F(e) = P_e(e)$.

Solution. Note that F is total so, by definition $f(e) = F(e) = P_e(e)$. But by the rule defined above for f, we have $f(e) = P_e(e) + 1$, a contradiction.

LO9. An instance of the decision problem `Zero` is a Gödel number x, and the problem is to decide if function ϕ_x equals the zero function, i.e. the function that outputs 0 on every input. Consider the function

$$g(x) = \begin{cases} 1 & \text{if } \phi_x(y) = 0 \text{ for all inputs y} \\ 0 & \text{otherwise} \end{cases}$$

(a) Evaluate $g(x)$ for each of the following Gödel number's x. Note: 2 out of 3 correct is considered passing.

i. $x = e_1$, where e_1 is the Gödel number of the program $P = T(1, 2), Z(1), S(2)$.

ii. $x = e_2$, where e_2 is the Gödel number of the program $P = T(1, 2), Z(1), Z(2), J(1, 2, 1)$.

iii. $x = e_3$, where e_3 is the Gödel number of the program that computes $g(x)$.

Hint: $T(m, n)$ means copy the value of R_m *to* R_n.

(b) Prove that $g(x)$ is not URM computable. In other words, there is no URM program that, on input x, always halts and either outputs 1 or 0, depending on whether or not ϕ_x equals the zero function. Do this by writing a program P that uses g and makes use of the `self` programming concept. Then show how P creates a contradiction.