CECS 329, Solutions to Learning Outcome Assessment 9 Problems, November 9th, Fall 2023, Dr. Ebert

Problems

LO5. A natural number $x \geq 1$ is said to be `perfect` iff the sum of all its divisors (that are less than x) add to x. For example, 28 is perfect since

$$1 + 2 + 4 + 7 + 14 = 28.$$

Using any of the primitive recursive functions defined in the Models of Computation lecture, show that the predicate function `perfect`(x) is primitive recursive, where `perfect`$(x) = 1$ iff x is perfect.

Solution. We have

$$\texttt{perfect}(x) = [(\sum_{z=1}^{x-1} \text{Div}(x, z) \cdot z) = x].$$

LO6. Solve the following problems.

(a) Compute the Gödel number for program $P = T(4,3), J(2,2,2), S(7), Z(4)$. Write your answer as a sum of powers of two minus 1 (see part b for an example). Show all work.

Solution. We have

$$\beta(T(4,3)) = 4\pi(3,2) + 2 = 4(39) + 2 = 158.$$

$$\beta(J(2,2,2)) = 4\xi(1,1,1) + 3 = 4\pi(\pi(1,1),1) + 3 = 4\pi(5,1) + 3 = 4(95) + 3 = 383,$$

$$\beta(S(7)) = 4(6) + 1 = 25,$$

and

$$\beta(Z(4)) = 4(3) = 12.$$

Thus,

$$\gamma(P) = \tau(158, 383, 25, 12) = 2^{158} + 2^{542} + 2^{568} + 2^{581} - 1.$$

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

$$2^9 + 2^{41} + 2^{50} + 2^{73} - 1.$$

Show all work.

Solution. $P = S(3), J(3,1,1), Z(3), T(2,2).$

LO7. Answer/solve the following.

(a) When simulating the computation $P_x(y)$, what is the only step in the simulation for which universal program P_U makes use of input y? Explain.

Solution. P_U only uses y to compute the initial configuration encoding σ_0.

(b) A universal program P_U is simulating a program that has 77 instructions and whose Gödel number is

$$x = 2^{31} + 2^{32} + 2^{43} + 2^{83} + 2^{128} + 2^{216} + \cdots + 2^{c_{77}} - 1.$$

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

$$\sigma = 2^5 + 2^9 + 2^{11} + 2^{14} - 1,$$

then provide the next configuration of the computation *and* its encoding.

Solution. We have

$$c = \tau^{-1}(\sigma) = (5, 3, 1, 2).$$

Also, $\beta(I_2) = 0$ and $0 \bmod 4 = 0$ implies that I_2 is the Zero instruction $Z(1)$, where $1 - 1 = 0/4 = 0$. Therefore,

$$c_{\text{next}} = (0, 3, 1, 3)$$

and

$$\tau(c_{\text{next}}) = 2^0 + 2^4 + 2^6 + 2^{10} - 1.$$

LO8. Answer the following.

(a) Define what it means to be a positive instance of decision problem `Self Accept`. Hint: "Gödel number x is a positive instance of `Self Accept` iff ______."

Solution. Gödel number x is a positive instance of `Self Accept` iff $P_x(x)\downarrow$, meaning P halts on its own Gödel number.

(b) The goal is to show that `Self Accept` is undecidable. We assume it is decidable by assuming that its characteristic function

$$f(x) = \begin{cases} 1 & \text{if } x \text{ is a positive instance of Self Accept} \\ 0 & \text{if } x \text{ is a negative instance of Self Accept} \end{cases}$$

is total computable. Provide the definition for how to compute the "antagonist" function $g(x)$ based on the value of $f(x)$.

Solution. See Section 3.1 of the Undecidability and Diagonalization Method lecture.

(c) By writing the values of $g(0), g(1), \ldots$ in the appropriate table cells (next to the number in that cell), verify that function g is different from each computable function ϕ_i, $i = 0, 1, \ldots$, Why does this create a contradiction?

Solution.

index\input x	0	1	2	⋯	i	⋯	**self accepting?**
$\phi_0(x)$	$\uparrow \to 0$	12	7	⋯	$\uparrow$	⋯	no
$\phi_1(x)$	8	$87 \to \uparrow$	36	⋯	96	⋯	yes
$\phi_2(x)$	7	5	$0 \to \uparrow$	⋯	$\uparrow$	⋯	yes
⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮
$\phi_i(x)$	0	32	65	⋯	$\uparrow \to 0$	⋯	no
⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮

On one hand g is different from every computable function (since we may assume that every computable function is listed in the table). However, g itself is computable which means it must be different from itself, a contradiction.

LO9. An instance of the decision problem `Infinite Range` is a Gödel number x, and the problem is to decide if function ϕ_x has an infinite range, meaning that ϕ_x has an infinite number of distinct outputs. Consider the function

$$g(x) = \begin{cases} 1 & \text{if } \phi_x \text{ has an infinite range} \\ 0 & \text{otherwise} \end{cases}$$

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

i. $x = e_1$, where e_1 is the Gödel number of the program that computes the function $\phi_{e_1}(y) = \lfloor \sqrt{y} \rfloor$.

Solution. $g(e_1) = 1$ since range$(\phi_{e_1}) = \mathcal{N}$ (every natural number is the square root of some natural number).

ii. $x = e_2$, where e_2 is the Gödel number of the program that computes the function $\phi_{e_2}(y) = e_2$.

Solution. $g(e_2) = 0$ since range$(\phi_{e_2}) = \{e_2\}$.

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

Solution. $g(e_3) = 0$ since range$(g) = \{0, 1\}$.

(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 has an infinite range. 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.

Solution.

Program P

Input y.

If $g(\text{self}) = 1$, then return 0.

Else return y.

Case 1: $g(\text{self}) = 1$. Then P should have an infinite range, but $\text{range}(P) = \{0\}$ which is finite, a contradiction.

Case 2: $g(\text{self}) = 0$. Then P should have a finite range, but $\text{range}(P) = \mathcal{N}$ since every natural number input is echoed as output. Thus P's range is infinite, a contradiction.