# CECS 329, Learning Outcome Assessment 10 Problems and Solutions, November 16th, Fall 2023, Dr. Ebert

## Problems and Solutions

LO6. Solve the following problems.

(a) Compute the Gödel number for program $P = S(5), Z(3), J(2,1,3), T(1,2)$. Write your answer as a sum of powers of two minus 1 (see part b for an example). Show all work.

**Solution.** We have

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

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

$$\beta(J(2,1,3)) = 4\xi(1,0,2) + 3 = 4\pi(\pi(1,0),1) + 3 = 4\pi(1,1) + 3 = 4(5) + 3 = 23,$$

and

$$\beta(T(1,2)) = 4\pi(0,1) + 2 = 4(2) + 2 = 10.$$

Thus,

$$\gamma(P) = \tau(17,8,23,10) = 2^{17} + 2^{26} + 2^{50} + 2^{61} - 1.$$

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

$$2^{22} + 2^{40} + 2^{84} + 2^{113} - 1.$$

Show all work.

**Solution.** $P = T(2,2), J(1,1,3), S(7), Z(8)$.

LO7. Answer/solve the following.

(a) When simulating the computation $P_x(y)$, why is it necessary for the universal program $P_U$ to compute the maximum register index used by program $P_x$?

**Solution.** $P_U$ needs to know the maximum index of any register used by $P_x$ since that number plus one gives the length of each configuration of the computation of $P_x(y)$. Thus, this number is needed to both encode and decode each configuration of the computation.

(b) A universal program $P_U$ is simulating a program that has 93 instructions and whose Gödel number is

$$x = 2^{29} + 2^{73} + 2^{117} + 2^{149} + 2^{168} + \cdots + 2^{c_{93}} - 1.$$

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

$$\sigma = 2^5 + 2^{13} + 2^{17} + 2^{18} + 2^{20} - 1,$$

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

**Solution.** We have

$$c = \tau^{-1}(\sigma) = (5, 7, 3, 0, 1).$$

Also, $\beta(I_1) = 29$ and $29 \bmod 4 = 1$ implies that $I_2$ is the Succcessor instruction $S(8)$, since $(29 - 1)/4 = 28/4 = 7$. Actually, this is an error in the problem, since $P_x$ has only four registers! Thus,

$$c_{\text{next}} = (5, 7, 3, 0, 2)$$

and

$$\tau(c_{\text{next}}) = 2^5 + 2^{13} + 2^{17} + 2^{18} + 2^{21} - 1,$$

LO8. Answer the following.

(a) One application of the diagonalization method is to prove that the set of partial unary functions cannot be placed in an infinite list, such as $f_0, f_1, f_2, \ldots$. This is done by assuming such a list of all functions exists and defining a partial unary function $g(x)$ that is different from every function in the list. Provide a definition for $g(x)$ that accomplishes this.

**Solution.** See Section 2.1 of the Undecidability and Diagonalization Method lecture.

(b) Using your definition from part a and 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 function $f_i$, $i = 0, 1, \ldots$. Conclude that the list does *not* include all partial unary functions.

**Solution.**

| index\input x | 0 | 1 | 2 | $\cdots$ | $i$ | $\cdots$ |
|---|---|---|---|---|---|---|
| $f_0(x)$ | $\uparrow \to 0$ | 12 | 7 | $\cdots$ | $\uparrow$ | $\cdots$ |
| $f_1(x)$ | 8 | $87 \to 88$ | 36 | $\cdots$ | 96 | $\cdots$ |
| $f_2(x)$ | 7 | 5 | $0 \to 1$ | $\cdots$ | $\uparrow$ | $\cdots$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $f_i(x)$ | 0 | 32 | 65 | $\cdots$ | $\uparrow \to 0$ | $\cdots$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ |

(c) By the diagonalization proof provided, it follows that the set of partial unary functions is too large to be placed in an infinite list. How does this prove that there are some partial unary functions that are *not* URM-computable?

**Solution.** Since the set of computable functions can be listed according to the index/Gödel number of each function, it follows that the above diagonalization procedure establishes a function $g$ that is not in this list, and thus is not computable. Note: the set of non-computable functions is thus *uncountable* (i.e. cannot be listed) since, otherwise, the set of all partial unary functions would occur in two lists (the computables and the non-computables) and these two lists could be interleaved into one list to give the set of all partiall unary functions, which contradicts the above fact that these functions cannot be listed.

LO9. An instance of the decision problem `Even Range` is a Gödel number $x$, and the problem is to decide if function $\phi_x$ has a non-empty range and whose members are (not necessarily all the) even numbers. Consider the function

$$g(x) = \begin{cases} 1 & \text{if } \phi_x \text{ has a non-empty range of even numbers} \\ 0 & \text{otherwise} \end{cases}$$

(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 $\phi_{e_1}(y) = 4y^2 + 3y$.

**Solution.** $g(e_1) = 0$ since $\phi_{e_1}(1) = 4 + 3 = 7$ which is odd, and so $\phi_{e_1}$ has odd numbers in its range.

ii. $x = e_2$, where $e_2$ is the Gödel number of the program that computes the function $\phi_{e_2}(y) = 6$.

**Solution.** $g(e_2) = 1$ since range($\phi_{e_2}$) $= \{6\}$ has only the even number 6.

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($g$) $= \{0, 1\}$ includes the odd number 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 $\phi_x$ has a non-empty even 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.**

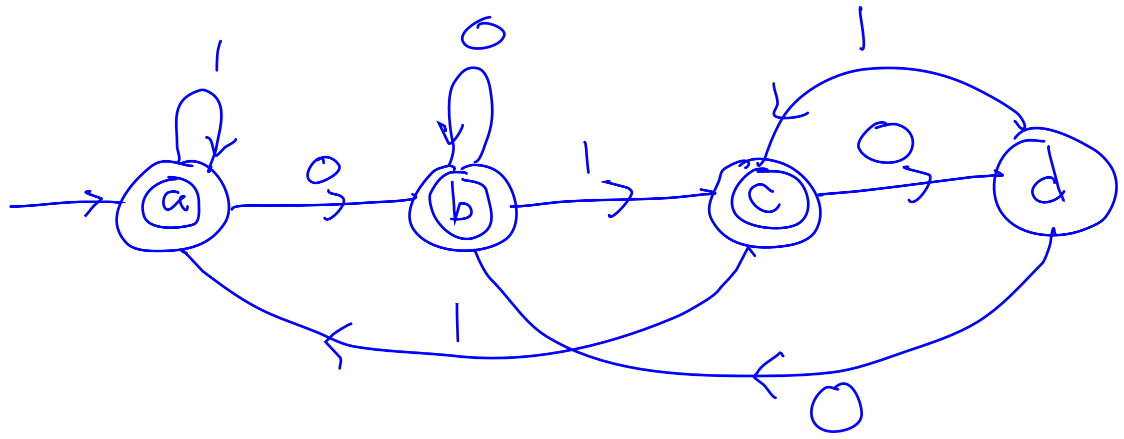
```
Program P
Input $y$.
If $g$(self) $= 1$, then return 1.
Else return 0.
```

Case 1: $g$(self) $= 1$. Then $P$ should have a range consisting of even numbers, but range($P$) $= \{1\}$ consists of a single odd number.

Case 2: $g$(self) $= 0$. Then $P$ should have at least one odd number in its range, but range($P$) $= \{0\}$, a contradiction.

LO10. Solve the following.

(a) Provide the state diagram for a DFA $M$ that accepts *all* binary words, *except* those that end with 010.



**Solution.**

(b) Show the computation of $M$ on inputs i) $w_1 = 110101$ and ii) $w_2 = 011010$.

$w_1 =$ 1 1 0 1 0 1

a a a b c d $\boxed{c} \Rightarrow$ accept

$w_2 =$ 0 1 1 0 1 0

a b c a b c $\boxed{d} \Rightarrow$ reject

**Solution.**