CECS 329, Learning Outcome Assessment 10, April 25th, Spring 2024, Dr. Ebert

Problems

LO6. Answer/solve the following.

(a) Compute the Gödel number of the program $P = J(2,1,2), T(3,2), Z(5), S(4)$.
Solution. $\beta(I_1) = 4\xi(1,0,1)+3 =$

$4\pi(\pi(1,0),1)+3 = 4\pi(1,1)+3 = 4\left[2^1(2(1)+1)-1\right]+3$

$= 23$

$\beta(T(3,2)) = 4\pi(2,1)+2 = 4(2^2(2(1)+1)-1)+2 = 46$

$\beta(Z(5)) = 16$. $\beta(S(4)) = (4)(3)+1 = 13$

$\therefore \gamma(P) = 2^{23}+2^{70}+2^{87}+2^{101}-1$

(b) A universal program P_U is simulating a program that has 56 instructions and whose Gödel number is

$$x = 2^{31} + 2^{78} + 2^{102} + 2^{182} + 2^{188} + \cdots + 2^{c_{56}} - 1.$$

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

$$\sigma = 2^2 + 2^4 + 2^9 + 2^{13} + 2^{16} - 1,$$

then provide the next configuration of the computation *and* its encoding.
Solution. $C = (2,1,4,3,2) \Rightarrow pc = 2$

$\beta(I_2) = 78-31-1 = 46 \Rightarrow I_2$ is a transfer instruction.

$(46-2)/4 = 11$ $\quad 11+1 = 12 = 2^2(2(1)+1) \Rightarrow T(3,2)$.

Hence, $C_{next} = (2,4,4,3,3)$ and

$\tau(C_{next}) = 2^2+2^7+2^{12}+2^{16}+2^{20}-1$

LO7. Answer the following.

(a) Describe what it means to be a positive instance of the `Self Accept` decision problem.
Solution. See Undecidability and Diagonalization Lecture Notes

(b) In applying the diagonalization method towards proving the undecidability of `Self Accept`, we defined a computable function $g(x)$ in terms of $f(x)$, where $f(x)$ is the decision function for `Self Accept`, which we assume to be total computable. Provide the formula for computing $g(x)$ and describe what needs to be established about $g(x)$ in order for the diagonalization method to be successful.

Solution. See Lecture Notes

(c) Suppose URM program P_{121} computes $\phi_{121}(x)$ and that

$$\phi_{121}(x) = \begin{cases} \lfloor\sqrt{x}\rfloor & \text{if } x \text{ is a perfect square} \\ \uparrow & \text{otherwise} \end{cases}$$

What is the value of $g(121)$ and how does this prove that $g \neq \phi_{121}$? Defend your answer.

Solution. $\phi_{121}(121) = \sqrt{121} = 11$

$g(121) = \uparrow$, $\therefore g(121) \neq \phi_{121}(121)$

so $g \neq \phi$

LO8. An instance of the decision problem `Finite Domain` is a Gödel number x, and the problem is to decide if program P_x has a finite domain set, meaning that the inputs it halts on forms a finite set. Note that $\emptyset$ is considered a finite set. Consider the function

$$g(x) = \begin{cases} 1 & \text{if } P_x \text{ has finite domain} \\ 0 & \text{otherwise} \end{cases}$$

(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 $\phi_{e_1}(x) = 7$.
Solution. $g(e_1) = 0$ since $W_{e_1} = \mathbb{N}$

ii. $x = e_2$, where e_2 is the Gödel number of the program that computes the function

$$\phi_{e_2}(x) = \begin{cases} 1 & \text{if } x \leq 500 \\ \uparrow & \text{otherwise} \end{cases}$$

Solution. $g(e_2) = 1$ since $W_{e_2} = \{0, 1, \ldots, 500\}$

iii. $x = e_3$, where e_3 is the Gödel number of the program $P = S(1), T(2,1), J(1,2,1)$.

Solution. $g(e_3) = 1$ since $W_{e_3} = \emptyset$ as P loops forever on all inputs.

(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 P_x has a finite domain. Do this by writing a program P that uses g and makes use of the `self` programming construct. Then use a proof by cases to show that P creates a contradiction.

Solution.
Program P
Input y
If g(self) = 1, Return 1. // Makes Dom(P) = $\mathbb{N}$
Else // g(self) = 0
loop forever // makes Domain(P) = $\emptyset$

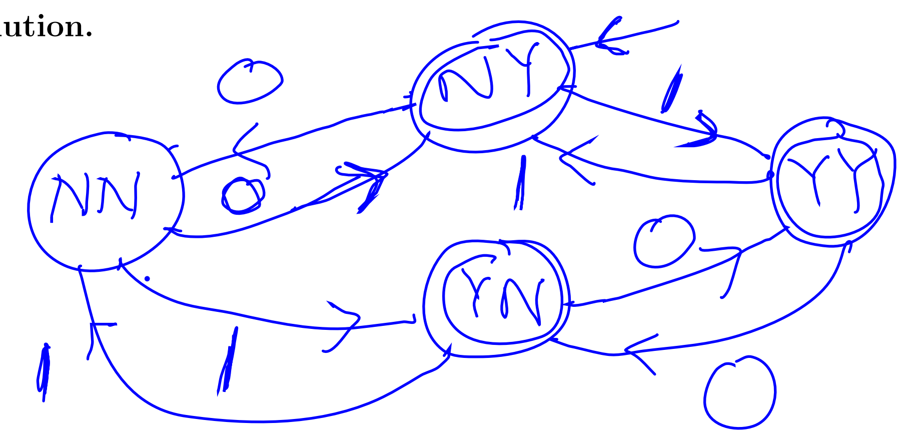
If g(self) = 1, then P has a finite domain. But P halts on all inputs.
If g(self) = 0, then P has infinite domain, but P loops forever on all inputs. Contradiction

Odd 1's? Even 0's?

LO9. Solve the following.

(a) Provide the state diagram for a DFA M that accepts a binary word w iff either w has an odd number of 1's or an even number of 0's (or both). Yes, zero is an even number.

Solution.



Accept

(b) Show the computation of M on inputs i) $w_1 = 10110111$ and ii) $w_2 = 0110110$.

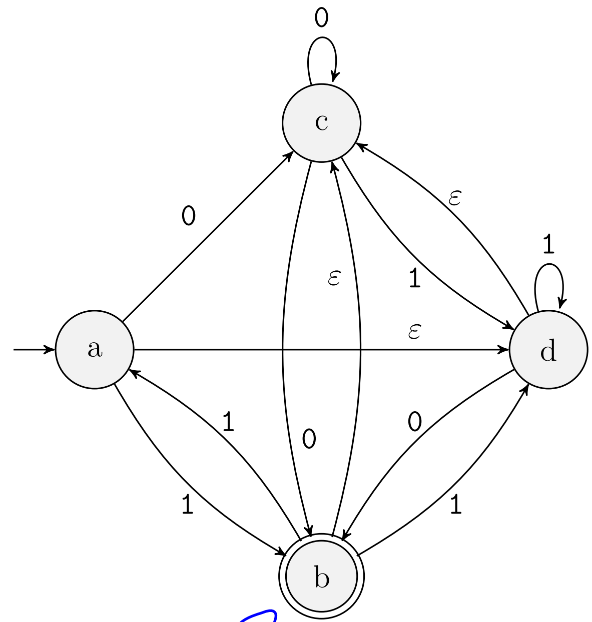
Solution.

w_1 = 1 NY 0 YY 1 YN 1 NN 0 YN 1 YY 1 NY 1 YY $\boxed{NY}$

w_2 = 0 NY 1 NN 1 YN 0 NN 1 NY 1 YY 0 NY $\boxed{NN}$

Reject

LO10. Do the following for the NFA N whose state diagram is shown below.



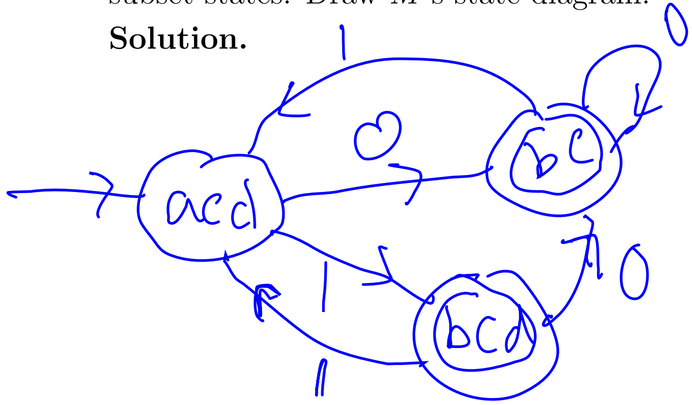
(a) Provide a table that represents N's δ transition function.

Solution.

$Q \backslash \Sigma$	0	1
a	{c}	{b,c}
b	∅	acd
c	{bc}	cd
d	bc	cd

(b) Use the table from part a to convert N to an equivalent DFA M using the method of subset states. Draw M's state diagram.

Solution.



(c) Show the computation of M on input $w = 11001$.

$acd \xrightarrow{1} bcd \xrightarrow{1} acd \xrightarrow{0} bc \xrightarrow{0} bc \xrightarrow{1} \boxed{acd}$

reject