CECS 329, Exam 2 Solutions, March 23rd, Spring 2023, Dr. Ebert

Rules for Completing the Problems

NO NOTES, BOOKS, ELECTRONIC DEVICES, OR INTERPERSONAL COMMUNICATION allowed when solving these problems. Make sure all these items are put away BEFORE looking at the problems. FAILURE TO ABIDE BY THESE RULES MAY RESULT IN A FINAL COURSE GRADE OF F.

Directions

Choose up to **six problems** to solve. Clearly mark each problem you want graded by placing an 'X' or check mark in the appropriate box. **If you don't mark any problems or mark more than six, then we will record grades for the six attempted problems that received the *fewest* points.**

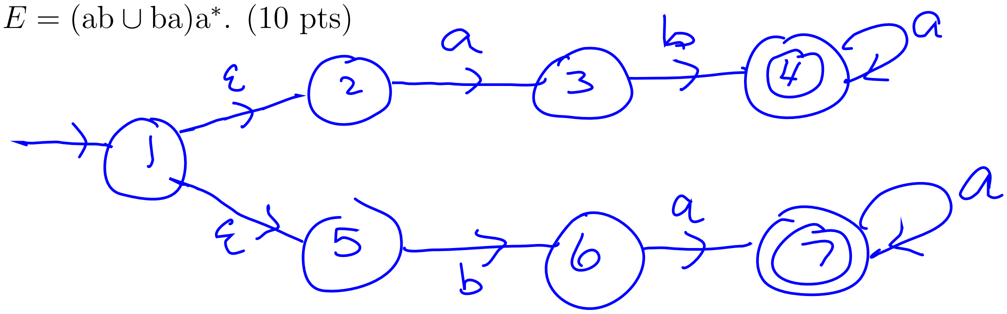
Problem	1	2	3	4	5	6	LO1	LO2	LO3	LO4	LO6
Grade?											
Result											

Your Full Name:

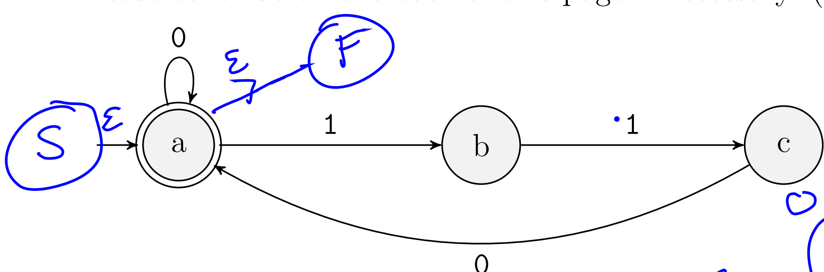
Class ID Number:

1. Solve each of the following.

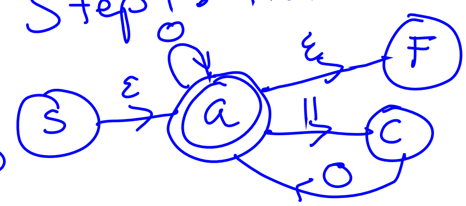
   a. Use the closure properties of NFA's to provide an NFA that accepts the language $L(E)$, where $E = (\mathrm{ab} \cup \mathrm{ba})\mathrm{a}^*$. (10 pts)



   b. Show each of the steps needed to convert the NFA shown below to a regular expression. Please continue on the back of this page if necessary. (15 pts)

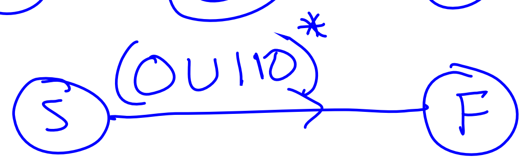


Step 1: Remove b



Step 2: Remove C

Step 3: Remove a



The above NFA accepts $(0 \cup 110)^*$.

2. Do the following.

   a. Provide the instructions for a URM that computes the function $f(x) = x^2$. Hint:

$$x^2 = \underbrace{x + x + \cdots + x}_{x \text{ times}}.$$

   Note: correctly solving this problem counts for passing LO5. (18 pts)

   Solution.

```
1:J(1,2,11)
2:S(2)
3:S(3)
4:J(1,3,6)
5:J(1,1,2)
6:S(4)
7:J(1,4,10)
8:Z(3)
9:J(1,1,2)
10:T(2,1)
```

   b. For each register used in your program, use one or more sentences to describe its purpose. (7 pts)

   Solution.

   - R1 stores input x.
   - R2 stores the program output.
   - R3 counts up to x and then resets.
   - R4 counts how many times R3 has counted up to x before being reset. Program continues to run until R4 itself counts up to x.

3. Let p_i denote the i th prime number, where $p_0 = 0$, $p_1 = 2$, $p_2 = 3$, etc.. Provide a *recursive* definition for p_i. You may use any PR functions from the General Models of Computation lecture examples and exercises, including predicate function $\text{Prime}(x)$ which evaluates to 1 iff input x is a prime number. Hint: use the fact that, for any prime number p, there must be another prime number that exceeds p and is no greater than $p! + 1$. Note: correctly solving this problem counts for passing LO5.

   Solution. See solution to Exercise 12.c in General Models of Computation lecture.

4. Do the following. Note: correctly solving this problem counts for passing LO7.

   a. A universal program P_U must be able to simulate any program, regardless of how many registers that program may use. It is able to accomplish this due to the fact that (5 pts)

      i. a universal program possesses a copy of each URM program's instructions and is able to look up the instructions based on the program's Gödel number x.

      ii. there are primitive recursive functions that allow for a universal program to perpetuate a simulation by extracting the necessary data from both the program encoding and the current-configuration encoding.

      iii. a universal program possesses an infinite number of registers that enable it to handle programs of any size.

      iv. All of the above are essential properties of P_U.

      Solution. ii.

   b. A universal program P_U is simulating a program whose Gödel number is

$$x = 2^{23} + 2^{33} + 2^{42} + 2^{53} + 2^{c_5} + \cdots + 2^{c_{76}} - 1.$$

      If the current configuration of $P_x(y)$ has encoding

$$\sigma = 2^3 + 2^6 + 2^{11} + 2^{19} + 2^{23} - 1,$$

      then provide the next configuration of the computation *and* its encoding. (15 pts)

      Solution. Current configuration: $(3, 2, 4, 7, 3)$. Program counter equals 3 and third instruction is encoded as $42 - 33 - 1 = 8$ which means it's $Z(3)$. So next configuration is $(3, 2, 0, 7, 4)$ and is encoded as

$$x = 2^3 + 2^6 + 2^7 + 2^{15} + 2^{20} - 1.$$

   c. Is your answer to part b a final configuration? Explain? (5 pts)

      Solution. No, since pc = 4 and pc > 76 is required for the program can halt.

5. Solve the following. Note: correctly solving either of these problems counts for passing LO5.

   a. Let x be a natural number, and let $c(x,i)$ denote the i th exponent in the power-of-two representation of x, where $x, i \geq 1$. For example, if $x = 35 = 2^5 + 2^1 + 2^0$, then $c(35,1) = 0$, $c(35,2) = 1$, $c(35,3) = 5$, and $c(55,i) = 0$, for $i \geq 4$. Show that $c(x,i)$ is primitive recursive. You may use any PR functions from the General Models of Computation lecture examples and exercises, including the function $\mathrm{bit}(x,i)$ which equals the i th bit of x when x is written as a binary number, and $\mathrm{len}(x)$ which gives the length of x when written as a binary number. Hint: you may assume $c(x,i) = 0$ in case either $x = 0$ or $i = 0$. (15 pts)

      Solution. We have

$$c(x,i) = \begin{cases} 0 & \text{if } i = 0 \vee i > \sum_{j=0}^{\mathrm{len}(x)} \mathrm{bit}(x,j) \\ \underset{z \leq \mathrm{len}(x)}{\lambda} \left(\sum_{j=0}^{z} \mathrm{bit}(x,j) = i\right) & \text{otherwise} \end{cases}$$

   b. Use the $c(x,i)$ function from part a to show that $\mathrm{IncrementComponent}(x,i)$ is primitive recursive, where x is the encoding of a tuple, and $1 \leq i \leq k(x)$ is a tuple component index. For example, suppose the tuple is $t = (3,4,1)$. Then its encoding is

$$x = \tau(t) = 2^3 + 2^8 + 2^{10} - 1 = 1287.$$

      Then $\mathrm{IncrementComponent}(1287,2) = 2567$ gives the encoding of the tuple $t' = (3, 4+1, 1) = (3,5,1)$ which is the result of incrementing the the 2nd component of t and encoding the result. (10 pts)

      Solution. We have

$$\mathrm{IncrementComponent}(x,i) = \sum_{j=1}^{i-1} 2^{c(x,j)} + \sum_{j=i}^{k(x)} 2^{c(x,j)+1}.$$

The following is an outline of a diagnonalization argument that proves that the following function is not computable:

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

In other words, $g(x) = 1$ iff the program having Gödel number x halts when its own Gödel number x serves as input, and it outputs 0 otherwise. We assume that g is a total URM-computable function and show a contradiction. Note: correctly solving this problem counts for passing LO8.

a. Define function $f(x)$ as

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

If URM program G has 500 instructions and computes $g(x)$, provide one or more additional URM instructions that, if appended to G, creates a new program F that correctly computes $f(x)$. Explain why this works. Hint: the first instruction appended to G should have assigned number 501. (10 pts)

Solution.

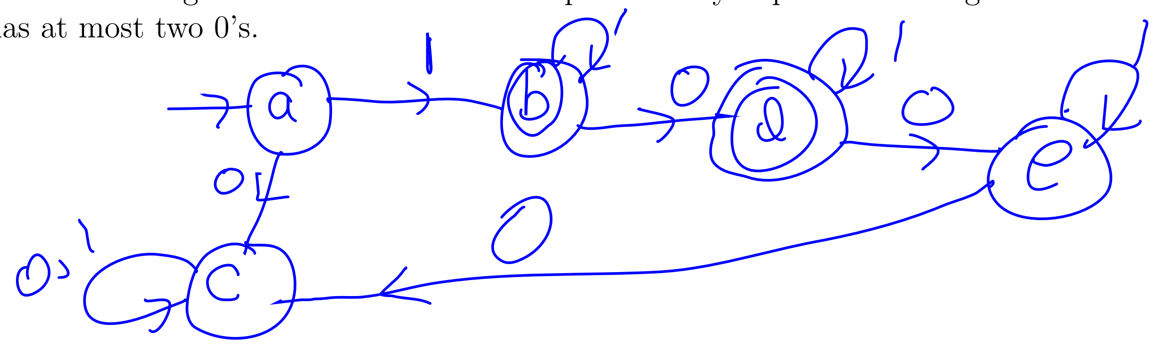
```
1-500. G[1:500]
501. Z(2)
502. J(1,2,504) //g(x) = 0
503. J(1,1,503) //g(x) = 1 so f(x) must be undefined via this infinite loop
504. S(1)  //f outputs 1 since g(x) = 0
```

b. Let e denote the Gödel number of program F from part a. In other words $F = P_e$. Show that a contradiction arises when we try to compute $F(e) = P_e(e)$. Analyze the two cases $F(e) = 1$ and $F(e) =\uparrow$, and show that each leads to a contradiction. (15 pts)

Solution. If $F(e) = 1$, then $g(e) = 0$ which means $F(e) \uparrow$, a contradiction. On the other hand, if $F(e) \uparrow$, then $g(x) = 1$ which means $F(e) \downarrow$, a contradiction.

LO1. Do the following.

(a) Provide the state diagram of a DFA M that accepts a binary w provided w begins with a 1 and has at most two 0's.



(b) Show the computation of M on input i) $w = 1101$ and ii) $w = 100101$.

1 1 0 1
a b b d d ⇒ accept

1 0 0 1 0 1
a b d e e c c ⇒ reject

LO3. Provide a regular expression E for which $L(E)$ denotes the set of words w for which w begins with a 1 and has at most two 0's.

Solution. We have

$$1^+ \cup 1^+01^* \cup 1^+01^*01^*.$$

LO4. Do the following.

(a) Provide a context-free grammar $G = (V, \Sigma, R, S)$ for which $L(G)$ consists of all binary words for which there are an equal number of 0's and 1's.

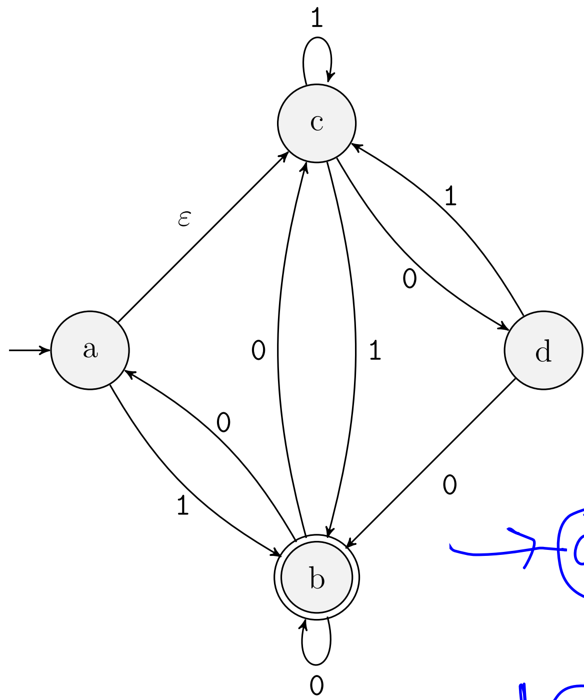
Solution. We have

$$S \to 0S1 \mid 1S0 \mid SS \mid \varepsilon.$$

(b) Use the grammar from part a to provide a leftmost derivation of the word 001110. **Solution.** We have

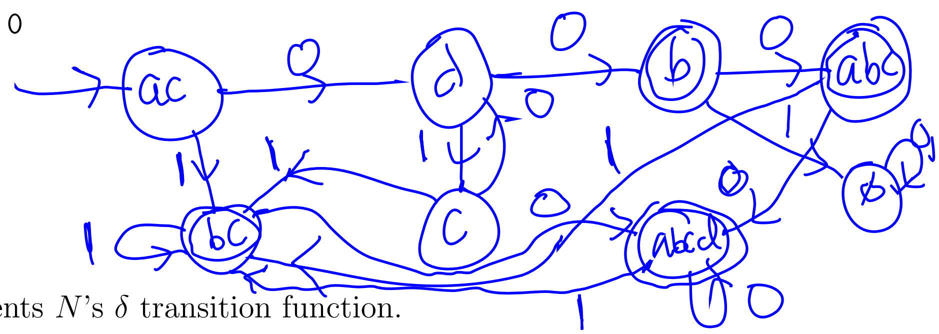
$$S \to SS \to 0S1S \to 00S11S \to 00111S0 \to 001110.$$

LO2. Do the following for the NFA N whose state diagram is shown below.



a)

δ \ Σ	0	1
a	$\emptyset$	b
b	abc	$\emptyset$
c	d	bc
d	b	c



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

(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.

(c) Show the computation of M on input $w = 11000$.

ac $\xrightarrow{1}$ bc $\xrightarrow{1}$ bc $\xrightarrow{0}$ abcd $\xrightarrow{0}$ abcd $\xrightarrow{0}$ abcd

$\Downarrow$

accept

LO6. Solve the following.

(a) Compute the Gödel number for program $P = T(3,2), J(1,2,3), Z(3), S(6)$. Write your answer as a sum of powers of two minus 1.

Solution.

$$\beta(T(3,2)) = 4(11) + 2 = 46.$$

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

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

$$\beta(S(6)) = 4(5) + 1 = 21.$$

$$\gamma(P) = \tau(46, 79, 8, 21) = 2^{46} + 2^{126} + 2^{135} + 2^{157} - 1.$$

(b) Provide the instructions of the program whose Gödel number is

$$x = 2^4 + 2^{14} + 2^{301} + 2^{381} - 1.$$

Solution. $\gamma^{-1}(x) = Z(2), S(3), T(4,5), J(1,2,3)$.