

MAD 3512 - THEORY OF ALGORITHMS

TEST #2 – FA 2022

FLORIDA INT'L UNIV.

TIME: 75 min.

*Answer all 6 questions. No calculators, notes, or on-line stuff are allowed. An unjustified answer will receive little or no credit. Draw a line to separate each of your 6 solutions to the 6 questions.*

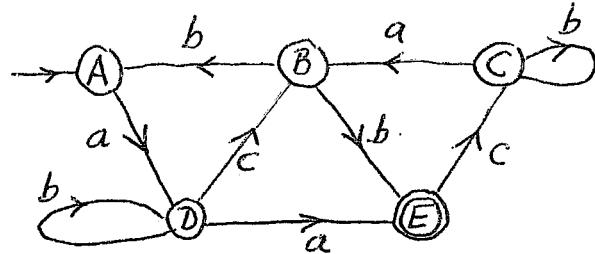
- (16) 1. (a) Find an NFA,  $M$ , which is *equivalent* to the RLG  $G$  given below.

$G: \rightarrow A, A \rightarrow 01A, A \rightarrow 1C, C \rightarrow 10, C \rightarrow \lambda, C \rightarrow 1D,$   
 $C \rightarrow B, B \rightarrow \lambda, B \rightarrow 0C, B \rightarrow 11, D \rightarrow 1A.$

- (b) Find an RLG,  $G$ , which is equivalent to the NFA in Problem 2(a) below.

- (16) 2. (a) Find a *regular expression* for the language accepted by the NFA  $M$  shown on the right.

- (b) Write down what the *Halting problem* says.



- (16) 3. (a) Write down the *initial functions* & define what " $F = \text{prec}(g, h)$ " means.

- (b) Show that  $F(x, y) = 4x + 5y + 2$  is a *primitive recursive function* on  $\mathbb{N} \times \mathbb{N}$  by finding primitive recursive functions  $g$  and  $h$  such that  $F = \text{prec}(g, h)$ .

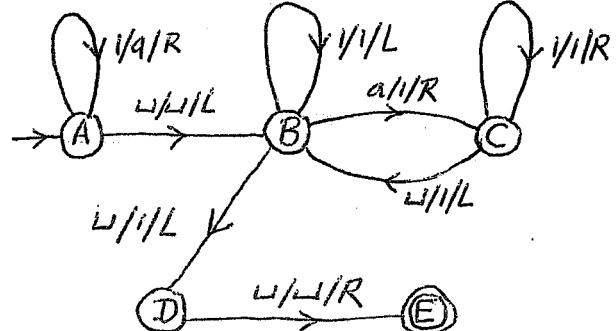
- (16) 4. (a) Define what "f is obtained by the *minimization* of the function  $g: \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ " means, and define what is a  $\mu$ -*recursive* partial function on  $\mathbb{N}^n$ .

- (b) Let  $f(x) = \text{Ceiling function of } [(x+2)^{2/3}]$ . Show that  $f$  is a  $\mu$ -*recursive* function.  
*[You may use the fact that PRED, MONUS, ADD, MULT & SIGN are primitive recursive if needed in #4, but you are not allowed to do so in Question #3.]*

- (18) 5. (a) Define what is a *Turing computable partial-function*  $f: \mathbb{N}^n \rightarrow \mathbb{N}$ .

- (b) Show what happens at each step if (i) 1 and (ii) 11 are the inputs for the TM,  $M$ , shown on the right.

- (c) What is the *function computed* by  $M$  in monadic (base 1) notation?

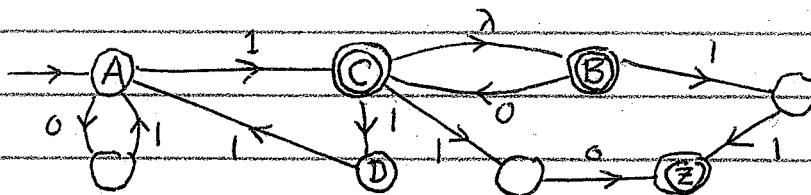


- (18) 6. Determine which of the following languages are regular and which are not.

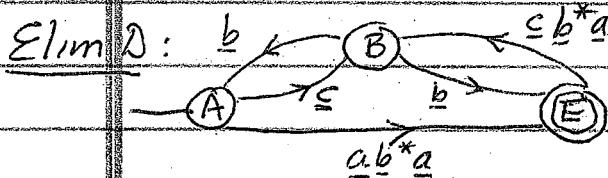
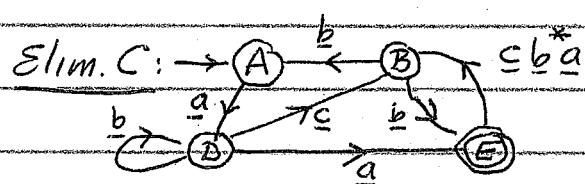
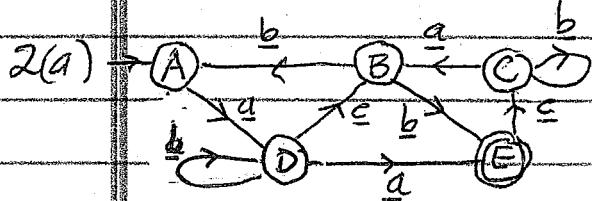
- (a)  $L_1 = \{a^k b^n : k \pmod 3 < (2 - n^2) \pmod 3\}$  (b)  $L_2 = \{b^k c^n : k > 2 + n^2\}$ .

*[If you say that it is regular, you must find a regular expression for it; if you say it is non-regular, you must give a complete proof.]*

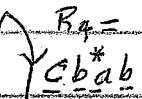
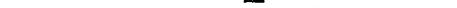
1(a) M:

(b) G:  $\rightarrow A, A \rightarrow aD, D \rightarrow bD, D \rightarrow cB, D \rightarrow aE$  $B \rightarrow bA, B \rightarrow bE, E \rightarrow cC, C \rightarrow bC, C \rightarrow aB, E \rightarrow \lambda$ 

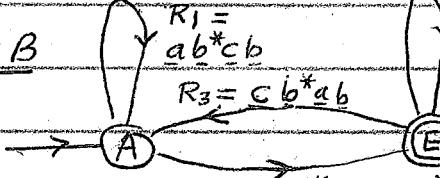
(9 transitions produces 9 prod. &amp; init. &amp; acc. st 2 more.)



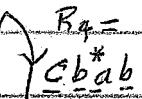
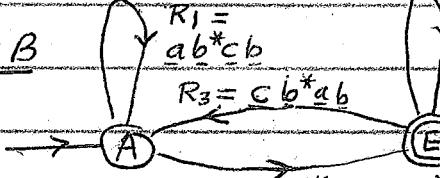
Elim. C:



Elim. D:



Elim. B:



$$\therefore L(M) = R_1^* R_2 (R_4 + R_3 R_1^* R_2)^*$$

$$= (\underline{ab^*cb})^* [\underline{ab^*(a+cb)}] \cdot [\underline{cb^*ab} + (\underline{cb^*ab}) \cdot (\underline{ab^*cb})^* \cdot \underline{ab^*(a+cb)}]^*$$

2(b) Halting Problem: Is there a TM, H, such that for an arb. TM M & an arb. input w for M; H halts in an accepting on  $c(M)\#c(w)$  when M halts in an acc. st. & H halts in a non-acc. st. on  $c(M)\#c(w)$  when M fails to halt on w? Here  $c(M)$  &  $c(w)$  are the codings of M & w into the input alphabet of H.

3(a) The initial functions are (i) the constant 0, (ii) the zero function of 1 var.  $Z(x) = 0$ , (iii) the successor function  $S(x) = x + 1$ , and (iv) the projective functions  $I_{k,n}(x_1, \dots, x_n) = x_k$  if  $1 \leq k \leq n$  and  $x$  if  $k = 0$ .

$F = \text{prec}(g, h)$  is the function  $F: N^{n+1} \rightarrow N$  defined by putting  $F(x, 0) = g(x)$  &  $F(x, s(y)) = h(x, y, F(x, y))$ . Here  $x = \langle x_1, x_2, \dots, x_n \rangle$ .

(2)

3(b) We have  $F(x, 0) = 4x + 5(0) + 2 = 4x + 2$ , [This is  $g(x)$ ]  
 and  $F(x, s(y)) = 4x + 5(y+1) + 2 = (4x + 5y + 2) + 5$ . This is  $h(x, y, F(x, y))$   
 So  $h = s_0 s_0 s_0 s_0 s_0 I_{3,3}$ . Also  $g(0) = 4(0) + 2 = 2$  &  $g(s(y)) = 4(y+1) + 2$   
 $= (4y + 2) + 4 = g(y) + 4$ . So  $g = \text{prec}(s_0 s_0 0, s_0 s_0 s_0 s_0 I_{2,2})$ . Thus  
 $F = \text{prec}(g, h) = \text{prec}(\text{prec}(s_0 s_0 0, s_0 s_0 s_0 s_0 I_{2,2}), s_0 s_0 s_0 s_0 s_0 I_{3,3})$

4(a) The partial function  $f: \mathbb{N}^n \rightarrow \mathbb{N}$  obtained by minimization of  $g$   
 is defined by  $f(x) = \begin{cases} \text{smallest value of } y \text{ such that } g(x, y) = 0 \\ \text{undefined, if } g(x, y) > 0 \text{ for each } y \in \mathbb{N}. \end{cases}$

A  $\mu$ -recursive function on  $\mathbb{N}$  is any partial function on  $\mathbb{N}^n$  that can be obtained from the initial functions by a finite no. of applications of composition, cartesian products, primitive recursion, and minimization on total functions.

(b) Let  $g(x, y) = (x+y)^2 - y^3$ . Then  $(\mu y)[g(x, y) = 0] = f(x)$ . So  
 $f = \mu[\text{MONUS} \circ \{\text{MULT} \circ (s_0 s_0 I_{1,2} \uparrow s_0 s_0 I_{1,2}) \wedge \text{MULT} \circ (I_{0,2} \wedge \text{MULT} \circ (I_{2,2} \wedge I_{2,2}))\}_0]$   
 and hence  $f$  is a  $\mu$ -recursive function.

5(a) A partial function  $f: \mathbb{N}^n \rightarrow \mathbb{N}$  is Turing-computable if we can find a TM  $M$  such that for an input  $x \in \text{dom}(f)$ ,  $M$  halts on  $x$  in an accepting state  $q_f$ , and outputs  $f(x)$ . If  $x \notin \text{dom}(f)$ , then  $M$  halts in a non-accepting state on  $x$  or fails to halt on  $x$ .

(b)  $\langle A, \underline{1} \rangle \vdash \langle A, a \underline{\omega} \rangle \vdash \langle B, \underline{a} \rangle \vdash \langle C, \underline{1} \underline{\omega} \rangle \vdash \langle B, \underline{1} \rangle$   
 $\vdash \langle B, \underline{\omega} \underline{1} \rangle \vdash \langle D, \underline{\omega} \underline{1} \underline{1} \rangle \vdash \langle E, \underline{1} \underline{1} \rangle$  halts  
 $\langle A, \underline{1} \rangle \vdash \langle A, a \underline{1} \rangle \vdash \langle A, aa \underline{\omega} \rangle \vdash \langle B, aa \rangle \vdash \langle C, a \underline{1} \underline{\omega} \rangle$   
 $\vdash \langle B, a \underline{1} \rangle \vdash \langle B, a \underline{1} \underline{1} \rangle \vdash \langle C, \underline{1} \underline{1} \rangle \vdash \langle C, \underline{1} \underline{1} \rangle$   
 $\vdash \langle C, \underline{1} \underline{1} \underline{\omega} \rangle \vdash \langle B, \underline{1} \underline{1} \underline{1} \rangle \vdash \langle B, \underline{1} \underline{1} \underline{1} \rangle \vdash \langle B, \underline{1} \underline{1} \underline{1} \rangle$   
 $\vdash \langle B, \underline{\omega} \underline{1} \underline{1} \underline{1} \rangle \vdash \langle D, \underline{\omega} \underline{1} \underline{1} \underline{1} \rangle \vdash \langle E, \underline{1} \underline{1} \underline{1} \rangle$  halts

(c) We have  $f(1) = 3$  &  $f(2) = 5$ . We can also do a quick check & see that  $f(0) = 1$ . So  $f(n) = 2n+1$ .

$\langle A, \underline{\omega} \rangle \vdash \langle B, \underline{\omega} \underline{\omega} \rangle \vdash \langle D, \underline{\omega} \underline{1} \underline{\omega} \rangle \vdash \langle E, \underline{1} \rangle$  halts.

(3)

6(a)  $L_1$  is a regular language - and we will find a reg. expr. for it.

If  $n \equiv 0 \pmod{3}$ , then  $2-n^2 \equiv 2-0 \equiv 2 \pmod{3}$ , so  $k \equiv 0 \text{ or } 1 \pmod{3}$ .

If  $n \equiv 1 \pmod{3}$ , then  $2-n^2 \equiv 2-1 \equiv 1 \pmod{3}$ , so  $k \equiv 0 \pmod{3}$ , only.

If  $n \equiv 2 \pmod{3}$ , then  $2-n^2 \equiv 2-4 \equiv 1 \pmod{3}$ , so  $k \equiv 0 \pmod{3}$  only.

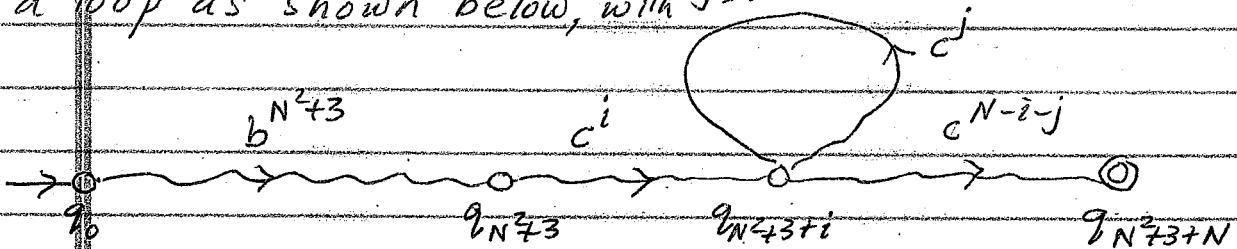
$\therefore (\lambda - a).(\underline{aaa})^*.(\underline{bbb})^* + \lambda.(\underline{aaa})^*.b.(\underline{bbb})^* + \lambda.(\underline{aaa})^*.bb.(\underline{bbb})^*$  is a regular expression which describes it.

contradiction

(b) We will prove that  $L_2$  is non-regular by using a proof by contradiction.

Suppose  $L_2$  was regular. Then we can find a  $\lambda$ -free NFA,  $M$ , with  $N$  states, such that  $\mathcal{L}(M) = L_2$ . Now  $b^{N^2+3}c^N \in L_2$  because  $N^2+3 > 2+(N)^2$ , so it will be accepted by  $M$ .

Since it takes  $N+1$  states to process  $c^N$  (and  $M$  has only  $N$  states) the acceptance track of  $b^{N^2+3}c^N$  must contain a loop as shown below, with  $j \geq 1$ .



Now if we ride this loop twice, we will see that  $M$

accepts the string  $b^{N^2+3} \cdot c^i \cdot c^j \cdot c^j \cdot c^{N-i-j} = b^{N^2+3} \cdot c^{N+j}$ .

But  $2+(N+j)^2 \geq 2+N^2+2Nj+j^2 \geq 2+N^2+2+1=N^2+5$ .

So  $N^2+3 \neq 2+(N+j)^2$ . Hence  $b^{N^2+3}c^{N+j} \notin L_2$ . But

this contradicts the fact that  $b^{N^2+3}c^{N+j} \in \mathcal{L}(M)$  &  $\mathcal{L}(M) = L_2$ .

Hence  $L_2$  must be a non-regular language.

END.