

Answer all 6 questions. No calculators, formula sheets or cell-phones are allowed. Provide all reasoning and show all working. An unjustified answer will receive little or no credit. Begin each of the 6 questions on 6 separate pages.

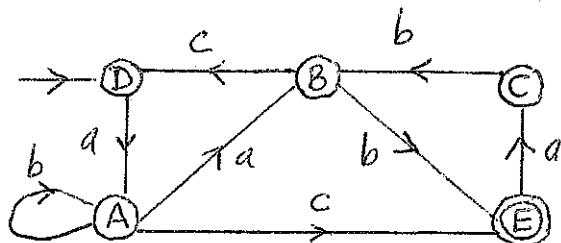
- (16) 1. (a) Find an NFA,  $M$ , which is equivalent to the RLG  $G$  given below.

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

- (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 asks.



- (16) 3. (a) Define the initial functions and the operation called primitive recursion.

- (b) Show that  $f(x,y) = 2x+3y+1$  is a primitive recursive function by finding primitive recursive functions  $g$  and  $h$  such that  $f = \text{prec}(g,h)$ .

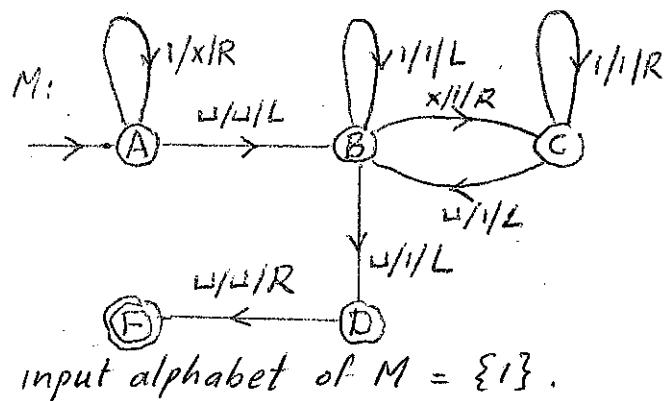
- (16) 4.(a) Define what it means for the function  $f$  (of  $n$  variables) to be obtained from the total function  $g$  (of  $n+1$  variables) by minimization.

- (b) Let  $f(x) = \text{Ceiling function of } [(2x+1)^{1/3}]$ . Show that  $f$  is a  $\mu$ -recursive function. [You may use the fact that PREP, 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 it means for a function  $f: D \rightarrow \mathbb{N}$  to be Turing-computable.

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

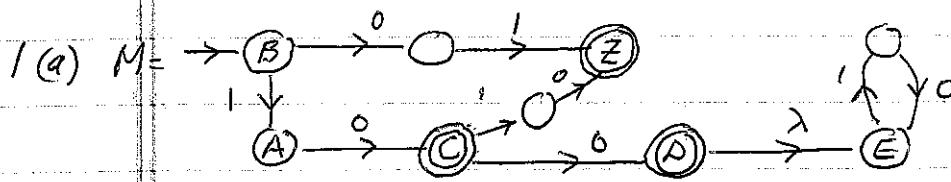
- (c) What is the function computed by  $M$  in monadic (base 1) notation?



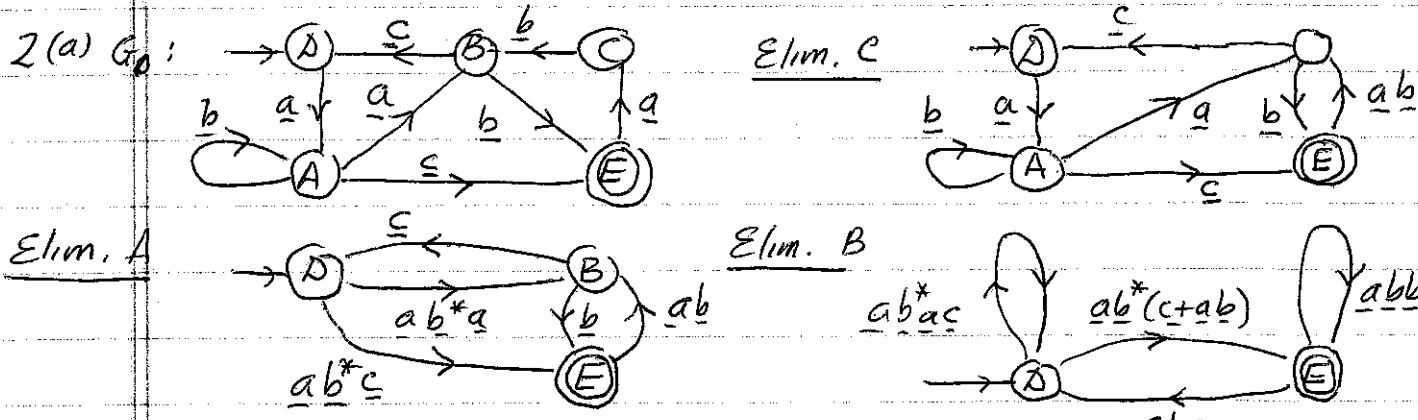
- (18) 6. Determine which of the following languages are regular and which are not.

$$(a) L_1 = \{a^k b^n : k \pmod{3} < (2n^2 - 1) \pmod{3}\} \quad (b) L_2 = \{b^k c^n : k < 2 + n^2\}.$$

[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.]



(b)  $G: \rightarrow D, D \rightarrow aA, A \rightarrow bA, A \rightarrow aB, A \rightarrow cE$   
 $B \rightarrow bE, E \rightarrow \lambda, E \rightarrow ac, C \rightarrow bb, B \rightarrow cd.$



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

$$= (\underline{ab}^* \underline{a} \underline{c})^* \cdot \underline{ab}^* (\underline{c} + \underline{ab}) (\underline{ab} \underline{b} + \underline{ab} \underline{c} \cdot (\underline{a} \underline{b} \underline{a} \underline{c})^* \cdot \underline{ab} (\underline{c} + \underline{ab}))^*$$

(b) Halting Problem: Is there a TM H such that for any TM M and any input  $w$ , when started on  $c(M)\#c(w) = M \& w$  coded as input for H, H will halt in an accepting state if M halts on  $w$ , and it will halt in a non-accepting state if M does not halt on  $w$ ?

3(a) The initial functions are (i) the constant 0, (ii) the zero function of one variable  $Z(x) = 0$ , (iii) the successor function  $S(x) = x+1$ , and (iv) the projective functions  $I_{k,n}(x_1, \dots, x_n) = x_k$  if  $1 \leq k \leq n$ , &  $\lambda$  if  $k=0$ .  
Primitive recursion is the operation that produces a function  $f: N^{n+1} \rightarrow N$  from the functions  $g: N^n \rightarrow N$  and  $h: N^{n+2} \rightarrow N$  by putting  $\begin{cases} f(\underline{x}, 0) = g(\underline{x}) \text{ where } \underline{x} = \langle x_1, \dots, x_n \rangle \\ f(\underline{x}, s(y)) = h(\underline{x}, y, f(\underline{x}, y)) \end{cases}$ .

(b) We will find primitive recursive functions  $g$  &  $h$  such that  $f = \text{prec}(g, h)$ . We have  $f(x, 0) = 2x + 3(0) + 1 = 2x + 1 \Leftarrow g(x)$  and  $f(x, s(y)) = 2x + 3(s(y) + 1) + 1 = (2x + 3y + 1) + 3 = f(x, y) + 3 \Leftarrow h(x, y, f(x, y))$ .

$\therefore g(x) = 2x+1 \Rightarrow g(y) = 2y+1$  &  $h = s_0 s_0 s_0 I_{3,3}$ . Now

$$g(0) = 1 \quad \& \quad g(s(y)) = 2(y+1)+1 = (2y+1)+2 = g(y)+2. \quad \text{So}$$

$$g = \text{prec}(s_0 0, s_0 s_0 I_{2,2}) \quad \therefore f = \text{prec}(\text{prec}(s_0 0, s_0 s_0 I_{2,2}), s_0 s_0 I_{3,3}).$$

4(a) The partial function  $f: N^N \rightarrow N$  is obtained from the total function

$g: N^{N^N} \rightarrow N$  by minimization by letting  $x = \langle x_1, \dots, x_n \rangle$  & putting

$$f(x) = \begin{cases} \text{smallest value of } y \text{ such that } g(x, y) = 0, \\ \text{undefined, if } g(x, y) > 0 \text{ for each } y \in N. \end{cases}$$

(b) We will find a function  $g: N^2 \rightarrow N$  such that  $f = \mu[g, 0]$ .

$$\text{Let } g(x, y) = (2x+1) - y^3. \text{ Then } (xy)[(2x+1) - y^3] = [(2x+1)^{1/3}] = f(x).$$

$$\text{So } f = \mu[\text{MONUS} \circ (\text{SO ADD} \circ [I_{1,2} \wedge I_{1,2}] \wedge \text{MULT} \circ [I_{2,2} \wedge \text{MULT} \circ [I_{2,2} \wedge I_{2,2}]]), 0].$$

5(a)  $f: D \rightarrow N$  is Turing-computable if we can find a TM M such that

$w \in D \Leftrightarrow \langle q_0, w \rangle \vdash^* \langle q_F, f(w) \rangle$  is a halted computation in M with  $q_F \in A(M)$

$$(b) (i) \langle A, \underline{\omega} \underline{\omega} \underline{\omega} \rangle \vdash \langle B, \underline{\omega} \underline{\omega} \omega \rangle \vdash \langle C, \underline{\omega} \underline{\omega} \underline{\omega} \rangle \vdash \langle D, \underline{\omega} \underline{\omega} \underline{\omega} \underline{\omega} \rangle$$

$$(ii) \langle A, \underline{\omega} \underline{\omega} \underline{\omega} \rangle \vdash \langle A, \underline{\omega} \underline{x} \underline{1} \rangle \vdash \langle A, \underline{\omega} \underline{x} \underline{x} \underline{\omega} \rangle \vdash \langle B, \underline{\omega} \underline{x} \underline{x} \underline{x} \rangle \vdash \langle C, \underline{\omega} \underline{x} \underline{x} \underline{x} \rangle$$

$$\vdash \langle B, \underline{\omega} \underline{x} \underline{x} \underline{1} \rangle \vdash \langle B, \underline{\omega} \underline{x} \underline{x} \underline{1} \rangle \vdash \langle C, \underline{\omega} \underline{x} \underline{x} \underline{1} \rangle \vdash \langle C, \underline{\omega} \underline{x} \underline{x} \underline{1} \rangle \vdash \langle C, \underline{\omega} \underline{x} \underline{x} \underline{x} \rangle$$

$$\vdash \langle B, \underline{\omega} \underline{x} \underline{x} \underline{x} \rangle \vdash \langle D, \underline{\omega} \underline{x} \underline{x} \underline{x} \underline{x} \rangle \vdash \langle F, \underline{x} \underline{x} \underline{x} \underline{x} \rangle$$

$$(c) f(n) = 2n+1 \text{ because } f(0)=1, f(2)=5 \text{ & } f(1)=3 \text{ (check this)}$$

$$6(a) n \equiv 0 \pmod{3} \Rightarrow k \pmod{3} < 2(0)^2 - 1 \equiv 2 \pmod{3} \Rightarrow k \equiv 0 \text{ or } 1 \pmod{3}$$

$$n \equiv 1 \pmod{3} \Rightarrow k \pmod{3} < 2(1)^2 - 1 \equiv 1 \pmod{3} \Rightarrow k \equiv 0 \pmod{3}$$

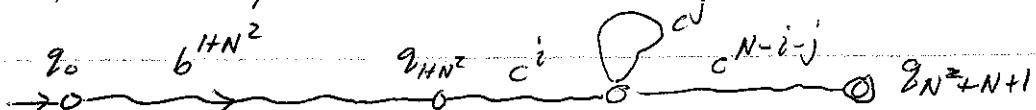
$$n \equiv 2 \pmod{3} \Rightarrow k \pmod{3} < 2(2)^2 - 1 \equiv 1 \pmod{3} \Rightarrow k \equiv 0 \pmod{3}.$$

$\therefore (\underline{a} + \underline{a})(\underline{aa}a)^* \cdot (\underline{bb}b)^* + (\underline{aa}a)^* \cdot (\underline{b}(\underline{bb}b)^* + \underline{bb} \cdot (\underline{bb}b)^*)$  is a reg. expr. that describes  $L_1$

(b) Suppose  $L_2$  was regular. Then we can find a  $\lambda$ -free NFA M (with N states)

such that  $L(M) = L_2$ . Since  $1+N^2 < 2+(N)^2$ ,  $b^{1+N^2} c^N \in L_2 = L(M)$ . Since it takes

$N+1$  states to process  $c^N$ , the acceptance track of  $b^{1+N^2} c^N$  must have loop as shown below.



Now if we skip the loop, we see that M will accept  $b^{1+N^2} \cdot c^i \cdot c^{N-i-j} = b^{1+N^2} c^{N-j}$ . But  $1+N^2 < 2+(N-j)^2$  because  $j \geq 1$ . So this contradicts the fact that  $L(M) = L_2$ . Hence  $L_2$  is not a regular language.