

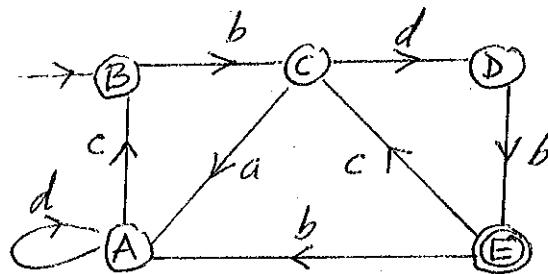
Answer all 6 questions. No calculators, formula sheets or cell-phones are allowed. Provide all reasoning and show all working. An unjustified answer will receive little or no credit. Begin each of the 6 questions on 6 separate pages.

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

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

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

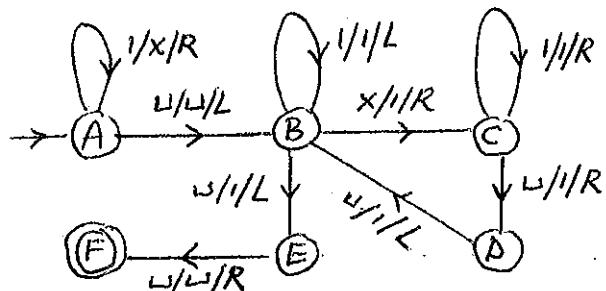
- (15) 2. (a) Find a regular expression for the language accepted by the NFA shown on the right.
(b) Write down the question that the Halting Problem asks & say if it is true or false.



- (20) 3. (a) Define what are the initial functions and what it means for f to be obtained from g and h by primitive recursion.
(b) Show that $f(x,y) = 2x+4y+3$ is a primitive recursive function by finding primitive recursive functions g and h such that $f = \text{prec}[g,h]$.

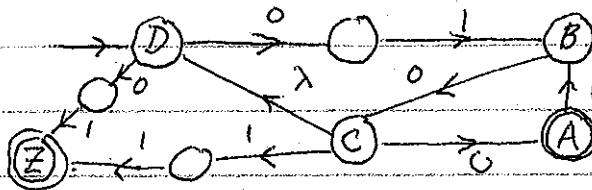
- (15) 4.(a) Explain completely how you can determine in a finite number of steps whether or not a given regular language L is infinite.
(b) If K , $K \cap L$, and $K^c \cap L^c$ are all regular languages; does it always follow that L must also be a regular language? (Justify your answer.)
[You may use any result proved in class to help you solve problem 4(b).]

- (15) 5. (a) Define when is a function from D to \mathbb{N} , Turing computable. Here $D \subseteq \mathbb{N}$.
(b) Show what happens at each step if (i) λ and (ii) 1 are the inputs for the TM, M , shown on the right.



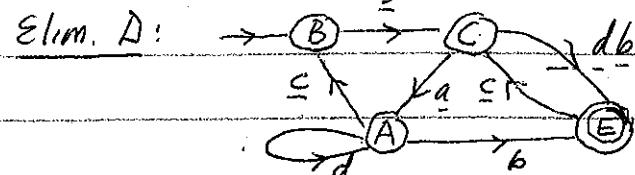
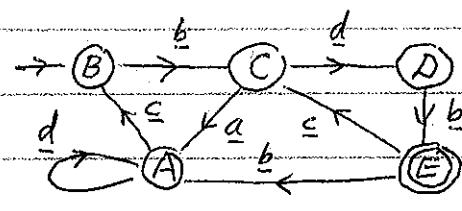
- (20) 6. Determine which of the following languages are regular and which are not.
(a) $L_1 = \{a^k.b^n : k \pmod 3 > n^2 - 2 \pmod 3\}$ (b) $L_2 = \{b^k.c^n : k < n^2 + 2\}$.
[If you say that it is regular, you must find a regular expression; if you say it is non-regular, you must give a complete proof.]

1(a)

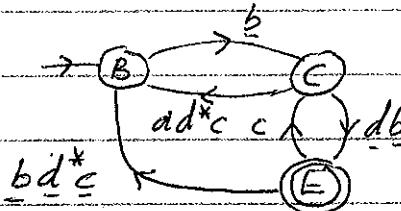


(b) $G: \rightarrow B, B \rightarrow bC, C \rightarrow aA$
 $C \rightarrow dB, A \rightarrow dA, A \rightarrow cB, D \rightarrow bE$
 $E \rightarrow bA, E \rightarrow cC, E \rightarrow \lambda$

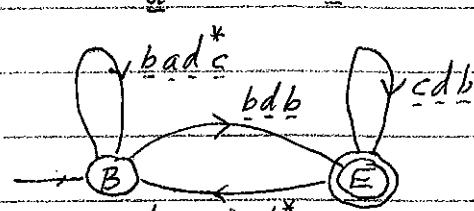
2(a)



Elim. A:



Elim. C:



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

$$= (\underline{bad}^*)^* \underline{bdb} (\underline{cdb} + (\underline{b+ca}) \underline{d}^* \underline{c}, (\underline{bad}^*)^* \underline{bdb})^*$$

(b) Halting Problem: Is there a TM H such that for any TM M & any input w for M, when started on $c(M) \# c(w)$, H will halt in an accepting state, if M halts on w; and H will halt in a non-accepting state if M doesn't halt on w? NO.

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

The function $f: \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ is obtained from $g: \mathbb{N}^n \rightarrow \mathbb{N}$ and $h: \mathbb{N}^{n+2} \rightarrow \mathbb{N}$ by primitive recursion if $f(x, 0) = g(x)$ and $f(x, s(y)) = h(x, y, f(x, y))$. Here $x = \langle x_1, \dots, x_n \rangle$.

(b) $f(x, y) = 2x + 4y + 3$. So $f(x, 0) = 2x + 3 \Rightarrow g(x) = 2x + 3$
and $f(x, s(y)) = 2x + 4(y+1) + 3 = (2x + 4y + 3) + 4 = f(x, y) + 4$

$$\therefore h(x, y, f(x, y)) = f(x, y) + 4. \text{ So } h = S \circ S \circ S \circ S \circ I_{3,3}.$$

$$g(0) = 3 = S \circ S \circ S \circ 0 \quad \& \quad g(s(y)) = 2(y+1) + 3 = (2y+3) + 2 = g(y) + 2.$$

$$\therefore g = \text{prec}(S \circ S \circ S \circ 0, S \circ S \circ I_{2,2}) \quad \& \quad f = \text{prec}(\text{prec}(S \circ S \circ S \circ 0, S \circ S \circ I_{2,2}), S \circ S \circ S \circ I_{3,3})$$

4(a) Since L is a regular language we can find a λ -free NFA M with $L(M) = L$. Let $N = \text{no. of states in } M$. Then L is infinite $\Leftrightarrow M$ accepts a string w with $N \leq |w| \leq 2N-1$. So try all the strings in the range N to $2N-1$ as inputs to M . If M accepts none of them, L will be finite; if M accepts at least one of them, L will be infinite.

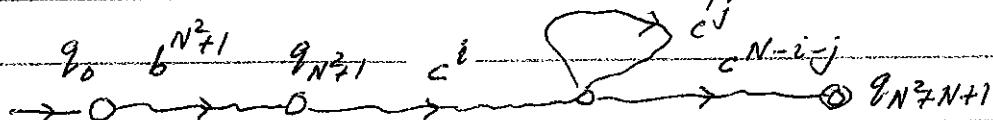
(b) Since $K \cap L^c$ is regular, $(K^c \cap L^c)^c = K \cup L$ will be regular by the Closure Theorem. Now $L = [(K \cup L) - K] \cup (K \cap L)$. Since K & $K \cap L$ are regular, L will therefore be regular by the Closure Thm.

5 (a) The function $f: D \rightarrow \mathbb{N}$ is Turing computable if we can find a TM M such that for each input $w \in \Sigma^*$, $\langle q_0, w \rangle \vdash \langle q_F, f(w) \rangle$ is a halted computation in M with $q_F \in A(M)$.

(b) (i) $\langle A, \sqcup \rangle \vdash \langle B, \sqcup \sqcup \rangle \vdash \langle E, \sqcup \sqcup \sqcup \rangle \vdash \langle F, \sqcup \sqcup \sqcup \rangle$
(ii) $\langle A, \sqcup \rangle \vdash \langle A, \times \sqcup \rangle \vdash \langle B, \times \rangle \vdash \langle C, \sqcup \sqcup \rangle \vdash \langle D, \sqcup \sqcup \rangle$
 $\vdash \langle B, \sqcup \sqcup \sqcup \rangle \vdash \langle B, \sqcup \sqcup \sqcup \rangle \vdash \langle B, \sqcup \sqcup \sqcup \rangle \vdash \langle E, \sqcup \sqcup \sqcup \rangle \vdash \langle F, \sqcup \sqcup \sqcup \rangle$

6 (a) $n \equiv 0 \pmod{3} \Rightarrow k \pmod{3} > 0^2 - 2 \equiv 1 \pmod{3} \Rightarrow k \equiv 2 \pmod{3}$
 $n \equiv 1 \pmod{3} \Rightarrow k \pmod{3} > 1^2 - 2 \equiv 2 \pmod{3} \Rightarrow \text{no value of } k$
 $n \equiv 2 \pmod{3} \Rightarrow k \pmod{3} > 2^2 - 2 \equiv 2 \pmod{3} \Rightarrow \text{no value of } k$
 L_1 is reg. because $a(aaa)^*(bbb)^*$ is a reg. expr. for L_1 .

(b) Suppose L_2 was regular. Then we can find a λ -free NFA M such that $L(M) = L_2$. Let N = no. of states in M . Since $b^{N^2}c^N \in L_2$, and it takes $N+1$ states to process c^N , the acceptance track of $b^{N^2}c^N$ must contain a loop, as shown below.



Now if we skip the loop, we see that M accepts $b^{N^2+1}c^{N-j}$. But $N^2+1 \not< (N-j)^2 + 2$. So this contradicts the fact that $L(M) = L_2$. Hence L_2 is non-regular. END