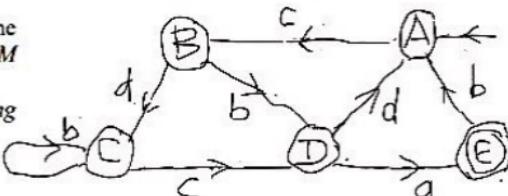
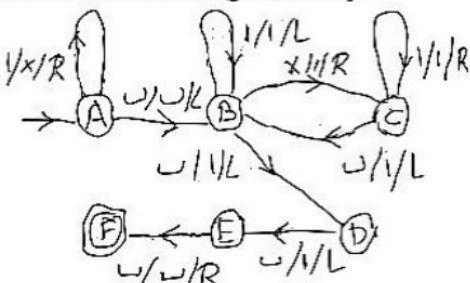


Answer all 6 questions. No calculators, formula sheets or cell-phones are allowed. Provide all reasoning and show all working. An unjustified answer will receive little or no credit. Begin each of the 6 questions on 6 separate pages.

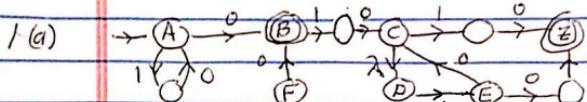
- (16) 1. (a) Find an NFA, M , which is equivalent to the RLG G given below.
 $G: A \rightarrow A, A \rightarrow 10A, A \rightarrow 0B, B \rightarrow 10C, B \rightarrow \lambda, C \rightarrow D,$
 $C \rightarrow 10, D \rightarrow 1E, E \rightarrow 0I, E \rightarrow 0C, F \rightarrow 0B.$
(b) Find an 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 $f(x,y) = 3x + 5y + 1$ is a primitive recursive function by finding primitive recursive functions g and h such that $f = \text{prec}(g,h)$.



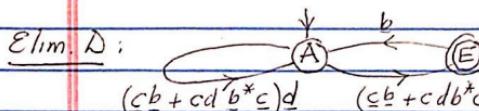
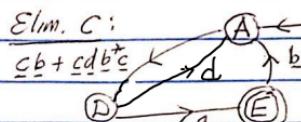
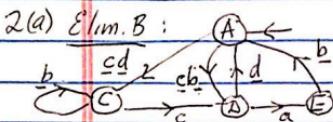
- (16) 4. (a) What's the difference between partial and μ -recursive functions on N .
(b) Let $f(x) = \lceil (x^2 + 2)^{1/3} \rceil$. Show that f is a μ -recursive function.
[You may use the fact that PRED, MONUS, ADD, MULT & SIGN are primitive recursive if needed in #4, but you are not allowed to do so in Question #3.]
- (18) 5. (a) Define what is a Turing-computable partial-function from N to 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) $L_1 = \{a^k b^n : k \pmod 3 < (2n^2 - 1) \pmod 3\}$ (b) $L_2 = \{b^k c^n : k > 2n^2 + 1\}$.
[If you say that the language is regular, then you must find a regular expression for it; and if you say it is non-regular, you must give a complete proof.]



(b) $G: \rightarrow A, A \rightarrow cB, B \rightarrow dC, C \rightarrow bC, C \rightarrow cD, \dots \begin{cases} 8 \text{ transitions} \\ 1 \text{ initial state} \\ 1 \text{ accepting state} \end{cases}$



$$R_1 = \underline{c}(b+db^*c)d$$

$$R_2 = \underline{c}(b+db^*c)a$$

$$R_3 = \underline{b}, R_4 = \emptyset$$

$$\therefore L(M) = R_1^* R_2 (R_4 + R_3 R_1^* R_2) = [c(b+db^*c)d]^* [c(b+db^*c)a]^* [b + b[c(b+db^*c)d]^* [c(b+db^*c)a]]^*$$

(b) Is there a TM H such that for an arb. TM M and an arb. input w for M , H will halt on $c(M)\#c(w)$ in an accepting state if M halts on w ; and H will halt on $c(M)\#c(w)$ in a non-accepting state if M does not halt on w ? [Here c codes the TM M and its input w into the input language of H .]

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

Prim. recursion is the operation that takes a function $g: N^n \rightarrow N$ & a function $h: N^{n+2} \rightarrow N$ and produces a function $f: N^{n+1} \rightarrow N$ by setting $f(x, 0) = g(x)$ and $f(x, s(y)) = h(x, y, f(x, y))$. Here $x = \langle x_1, x_2, \dots, x_n \rangle$.

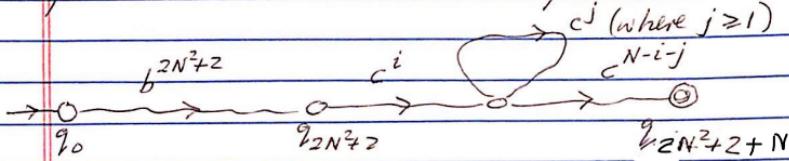
~~5(b) $f(x, y) = 3x + 5y + 1$. So $f(x, 0) = 3x + 1 \leftarrow g(x)$~~
 and $f(x, s(y)) = 3x + 5(y+1) + 1 = (3x + 5y + 1) + 5 = f(x, y) + 5$
 $\therefore f = \text{prec}(g, h)$ where $g(x) = 3x + 1$
 and $h = s_0 s_0 s_0 s_0 s_0 I_{3,3}$. Now $g(0) = 3(0) + 1 = 1$ &
 $g(s(y)) = 3(y+1) + 1 = (3y + 1) + 3$. $\therefore g = \text{prec}(s_0 0, s_0 s_0 s_0 I_{2,2})$
 $\therefore f = \text{prec}(g, h) = \text{prec}(\text{prec}(s_0 0, s_0 s_0 s_0 I_{2,2}), s_0 s_0 s_0 s_0 s_0 I_{3,3})$
 and so is a primitive recursive function.

~~4(a) A partial function $f: \mathbb{N} \rightarrow \mathbb{N}$ is allowed to be undefined for some values $n \in \mathbb{N}$. A μ -recursive function is a partial function that can be obtained from the initial functions using the operations of cartesian product, composition, primitive recursion & minimization on total functions. So the difference is that a μ -recursive function is a more restricted kind of partial func.~~
~~(b) Let $g(x, y) = (x^2 + 2) - y^3$. Then $f(x) = \{\bar{y} | g(x, y) = 0\}$~~
~~So $f_1 = \mu[g, 0] = \mu[\text{MONUS} \circ (s_0 s_0 \text{MULT} \circ [I_{1,2} \wedge I_{1,2}], \text{MULT} \circ [I_{2,2} \wedge \text{MULT} \circ [I_{2,2} \wedge I_{2,2}], 0)]$~~
~~and is thus μ -recursive.~~

~~5(a) A partial function $f: \mathbb{N} \rightarrow \mathbb{N}$ is Turing computable if we can find a TM M such that M halts in an accepting state with output $f(n) \Leftrightarrow n \in \text{dom}(f)$.~~
~~(b) (i) $\vdash \langle A, \underline{!} \rangle \vdash \langle A, x \sqsubseteq \rangle \vdash \langle B, x \sqcup \rangle \vdash \langle C, 1 \sqcup \rangle \vdash \langle B, 11 \rangle$
 $\vdash \langle B, \sqsubseteq 11 \rangle \vdash \langle D, \sqsubseteq 111 \rangle \vdash \langle E, \sqsubseteq 1111 \rangle \vdash \langle F, \sqsubseteq 1111 \rangle \text{ Halt}$~~
~~(ii) $\vdash \langle A, \underline{11} \rangle \vdash \langle A, x \underline{1} \rangle \vdash \langle A, x \times \underline{1} \rangle \vdash \langle B, x \times \underline{1} \rangle \vdash \langle C, x \underline{1} \sqcup \rangle$
 $\vdash \langle B, x \times 11 \rangle \vdash \langle B, x \underline{11} \rangle \vdash \langle C, 1 \underline{11} \rangle \vdash \langle C, 11 \underline{1} \rangle \vdash \langle C, 111 \underline{1} \rangle$
 $\vdash \langle B, 111 \underline{1} \rangle \vdash \langle B, 1111 \rangle \vdash \langle B, 111 \underline{1} \rangle \vdash \langle B, 1 \underline{111} \rangle \vdash \langle D, \underline{1111} \rangle$
 $\vdash \langle E, \underline{1111} \rangle \vdash \langle F, \underline{1111} \rangle \text{ Halt}$~~
~~(c) So $f(1) = 4$, $f(2) = 6$ & you can check that $f(0) = 2$
 $\therefore f(n) = 2n + 2$.~~

6(a) If $n \equiv 0 \pmod{3}$, then $k \pmod{3} < 2(0)^2 - 1 \equiv 2 \pmod{3}$; $k \equiv 0 \pmod{1}$
 If $n \equiv 1 \pmod{3}$, then $k \pmod{3} < 2(1)^2 - 1 \equiv 1 \pmod{3}$; $k \equiv 0 \pmod{3}$
 If $n \equiv 2 \pmod{3}$, then $k \pmod{3} < 2(2)^2 - 1 \equiv 1 \pmod{3}$; $k \equiv 0 \pmod{3}$
 $\therefore (\underline{a+a})(\underline{aa})^*(\underline{bb})^* + (\underline{aa})^*\underline{b}(\underline{bb})^* + (\underline{aa})^*\underline{bb}(\underline{bb})^*$ is a regular expr which describes L_1 . So L_1 is regular.

(b) Suppose L_2 was regular. Then we can find a λ -free NFA M_2 with N states such that $\delta(M_2) = L_2$. Now consider the string $b^{2N^2+2}c^N$. Since $\underbrace{2N^2+2}_k > \underbrace{2(N)^2+1}_n$, $b^{2N^2+2}c^N$ will be accepted by M_2 . Also since it takes $N+1$ states to process the c^N , the acceptance track must have a loop as shown below.



Now if we ride this loop twice, then we will see that M_2 accepts $b^{2N^2+2}c^i c^{2j} c^{N-i-j} = b^{2N^2+2}c^{N+j}$. But $2N^2+2 \neq 2(N+j)^2+1 = 2N^2+1+4Nj+2j^2$ because $j \geq 1$. Hence M_2 accepts a string which is not in L_2 , contradicting $\delta(M_2) = L_2$. Hence L_2 cannot be regular. So L_2 is non-regular.
 END.