

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.

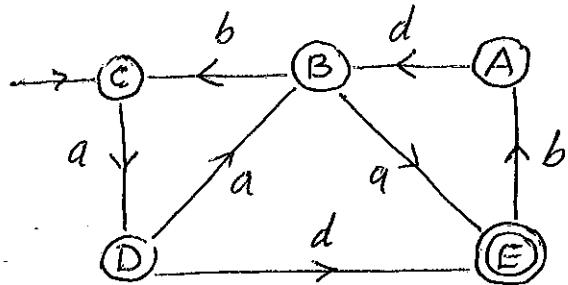
- (15) 1 (a) Find an NFA M which is equivalent to the RLG G given below.

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

- (b) Suppose F is a finite language and $L \cup F$ is a regular language. Is it always true that L must also be a regular language? (Justify your answer.)

- (15) 2 (a) Write down what the Halting Problem says.

- (b) Find a regular expression for the language accepted by the NFA shown on the right.



- (15) 3 (a) Define what it means for f to be obtained from g and h by primitive recursion.

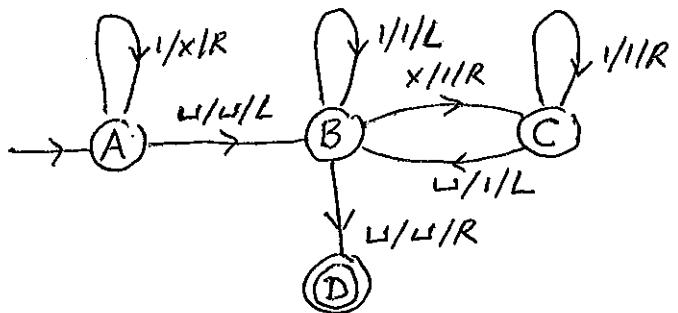
- (b) Show that $f(x,y) = 3x+2y+1$ is a primitive recursive function by finding primitive recursive functions g and h such that $f = \text{prec}[g,h]$.

- (20) 4 (a) Define what it means for the function $f: \mathbb{N}^n \rightarrow \mathbb{N}$ to be obtained from the total function $g: \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ by minimization.

- (b) Show that the functions $f(x) = \lceil (x/3) \rceil$ and $h(x) = x \pmod{2}$ are μ -recursive functions. [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 #3.]

- (15) 5 (a) Define what is a Turing-decidable binary relation R on \mathbb{N} .

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

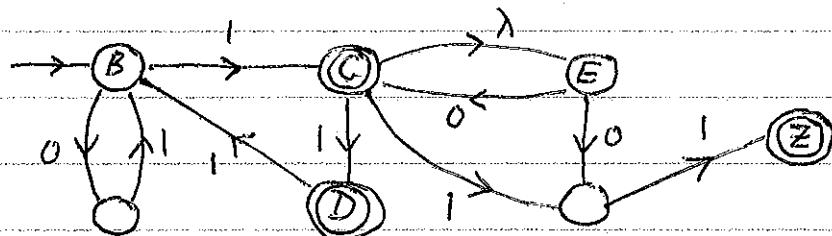


- (20) 6. Determine which of the following languages are regular and which are not.

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

[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.]

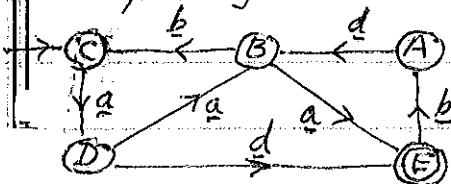
1(a)



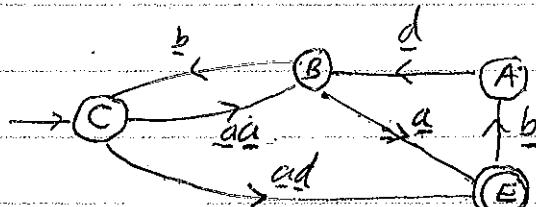
(b) YES. First observe that $F \cap L$ is finite because $F \cap L \subseteq F$. Now all finite languages are regular. So $F \cap L$ will be regular. Since $L \cup F$ is regular $(L \cup F) - F$ will also be regular. But $L = [(L \cup F) - F] \cup (L \cap F)$. Hence L is a union of regular languages and so is regular.

2(a) The Halting Problem is the question. Is there a TM H such that for any TM M and any input w for M , H will be in an accepting state if M halts on w & H will halt in a non-accepting state if M does not halt on w ?

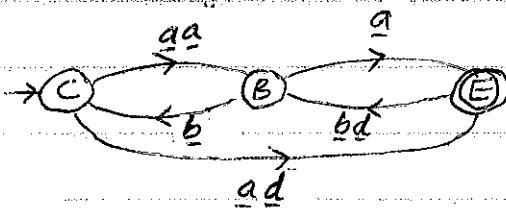
(b) Corresponding GFA



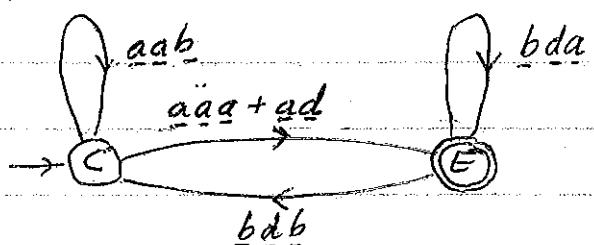
Eliminate D



Eliminate A



Eliminate B



$$R_1 = \underline{aab}, R_2 = \underline{aa} + ad = a(\underline{aa} + d), R_3 = \underline{bdb}, R_4 = bda$$

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

$$= (\underline{aab})^* a(\underline{aa} + d) (bda + \underline{bdb}(\underline{aab})^* a(\underline{aa} + d))^*$$

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

(b) $f(x, y) = 3x + 2y + 1$. So $f(x, 0) = 3x + 1 \Rightarrow g(x) = 3x + 1$
Also $f(x, s(y)) = 3x + 2(y+1) + 1 = (3x + 2y + 1) + 2 = f(x, y) + 2$
So $h(x, y, f(x, y)) = f(x, y) + 2$. $\therefore h = s_0 s_0 I_{3,3}$.
Now g is not yet written as a primitive recursive function
 $g(y) = 3y + 1$. So $g(0) = 1$ and $g(s(y)) = 3(y+1) + 1 = g(y) + 3$
 $\therefore g = \text{prec}[s_0 0, s_0 s_0 s_0 I_{2,2}]$. Hence $f = \text{prec}[g, h]$
 $= \text{prec}[\text{prec}[s_0 0, s_0 s_0 s_0 I_{2,2}], s_0 s_0 I_{3,3}]$. Thus f is
a primitive recursive function.

4(a) f is obtained from g by minimization if $\underline{x} = \langle x_1, \dots, x_n \rangle$ &
 $f(\underline{x}) = \begin{cases} \text{smallest } y \text{ such that } g(\underline{x}, y) = 0 \\ \text{undefined, when } g(\underline{x}, y) \geq 1 \text{ for each } y \in \mathbb{N}. \end{cases}$

(b) Let $g(x, y) = x - 3y$. Then $(\mu y)[g(x, y) = 0] = (\mu y)[x - 3y = 0]$
 $= \lceil x/3 \rceil = f(x)$. So $f = \mu[g, 0]$.
 $\therefore f = \mu[\text{MONUS} \circ (I_{1,2} \wedge \text{MULT} \circ ((s_0 s_0 s_0 z \circ I_{2,2}) \wedge I_{2,2}), 0)]$
 $\therefore f$ is a μ -recursive function.

(c) First observe that $h(x) = x \pmod{2} = \begin{cases} 0 & \text{if } x \text{ is even,} \\ 1 & \text{if } x \text{ is odd.} \end{cases}$

Now $(\mu y)[x - 2y = 0] = \lceil x/2 \rceil = k(x)$, say. So

$$k = \mu[\text{MONUS} \circ (I_{1,2} \wedge \text{MULT} \circ ((s_0 s_0 z \circ I_{2,2}) \wedge I_{2,2})), 0].$$

$$\text{But } h(x) = 2(\lceil x/2 \rceil) - x = \text{MONUS}(\text{MULT}(s(s(z(x))), k(x)), I_1(x))$$

$$\therefore h = \text{MONUS} \circ (\text{MULT} \circ ((s_0 s_0 z) \wedge k) \wedge I_1)$$

Since k is μ -recursive, it follows that h is μ -recursive.

5(a) The binary relation R on \mathbb{N} is Turing-decidable if we can find
a TM M such that for the input $\langle m, n \rangle$, M will halt in an

5(a) accepting state if $\langle m, n \rangle \in R$; and M will halt in a non-accepting state if $\langle m, n \rangle \notin R$.

$$(i) \langle A, \sqcup \rangle \vdash \langle B, \sqcup \sqcup \rangle \vdash \langle D, \sqcup \sqcup \sqcup \rangle$$

$$(ii) \langle A, \sqcap \rangle \vdash \langle A, \times \sqcup \rangle \vdash \langle B, \times \sqcup \sqcup \rangle \vdash \langle C, \sqcap \sqcup \rangle \\ \vdash \langle B, \sqcap \sqcap \rangle \vdash \langle B, \sqcup \sqcap \sqcup \rangle \vdash \langle D, \sqcap \sqcap \sqcap \rangle$$

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

$$n \equiv 1 \pmod{3} \Rightarrow k \pmod{3} > 1^2 \pmod{3} \Rightarrow k \equiv 2 \pmod{3}$$

$$n \equiv 2 \pmod{3} \Rightarrow k \pmod{3} > 2^2 \pmod{3} = 1 \pmod{3} \Rightarrow k \equiv 2 \pmod{3}$$

$$\therefore L_1 = (\underline{a+a}) \cdot (\underline{aaa})^* (\underline{bbb})^* + \underline{aa} \cdot (\underline{aaa})^* (\underline{bbb})^* b + \underline{aa} (\underline{aaa})^* (\underline{bbb})^* \underline{bb}$$

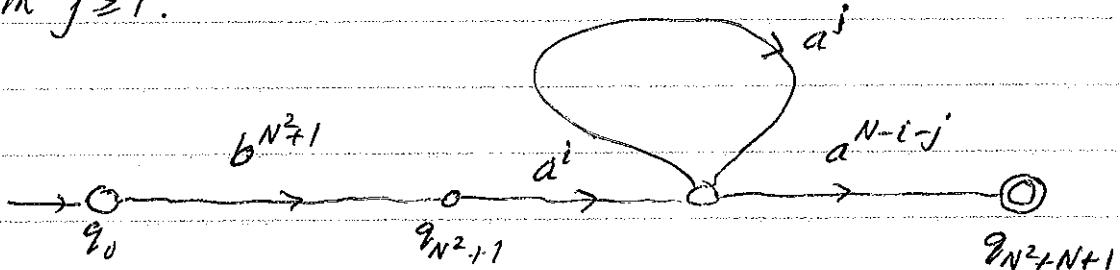
and so is a regular language.

(b) $L_2 = \{b^k a^n : k > n^2\}$. Suppose L_2 was regular. Then we can

find a λ -free NFA M such that $L(M) = L_2$. Let N = the number of states in M. Then $b^{N^2+1} a^N \in L_2$ because if we put $k = N^2+1$

& $n = N$, we will see that $k > n^2$. So M will accept $b^{N^2+1} a^N$.

Since it takes $N+1$ states to process the a^N , any acceptance track of $b^{N^2+1} a^N$ 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 $b^{N^2+1} a^i a^j a^j a^{N-i-j} = b^{N^2+1} a^{N+j}$. But $b^{N^2+1} a^{N+j} \notin L_2$ because $N^2+1 \neq (N+j)^2$. So this contradicts the fact that $L(M) = L_2$. Hence L_2 is non-regular.