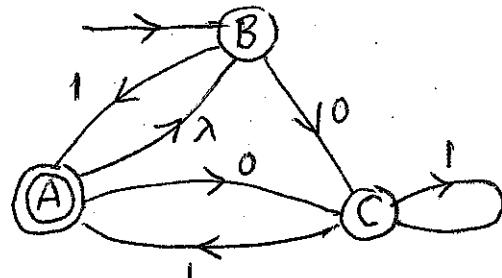


Answer all 6 questions. No calculators, formula sheets, or cellphones are allowed. Provide all reasoning and show all working. An unjustified answer will receive little or no credit. BEGIN EACH OF THE 6 QUESTIONS ON A SEPARATE PAGE.

- (15) 1.(a) Define what is a *regular expression* over the alphabet $V = \{a,b,c,d\}$
 (b) Let M be the *NFA* on the right. Find a *DFA* M_D which is *equivalent* to M .



- (15) 2. Find *regular expressions* which describe the languages below.
 (a) $L_1 = \{\varphi \in \{a,b\}^*: \varphi \text{ contains both } bab \text{ and } aab \text{ as substrings}\}$
 (b) $L_2 = \{\varphi \in \{0,1\}^*: \varphi \text{ contains at most one occurrence of the string } 11\}$.

- (20) 3. (a) Define what it means for two states p & q to be *indistinguishable* in a *DFA* M .
 (b) Partition the states of the *DFA*, M below into *blocks of indistinguishable states* and then find the equivalent *reduced machine*, M_R .

| | $\rightarrow A$ | \boxed{B} | \boxed{C} | D | E | \boxed{F} | G |
|---|-----------------|-------------|-------------|---|---|-------------|---|
| 0 | B | G | D | B | B | A | F |
| 1 | F | B | F | C | G | C | E |

- (15) 4. (a) Let $f(\omega) = [2.n_a(\omega) - n_b(\omega) - 3] \pmod{5}$. Find a *DFA*, M which accepts the language, $L_4 = \{\varphi \in \{a,b\}^*: f(\varphi) \text{ is odd}\}$.
 (b) If $\varphi = ababa$ find $f(\varphi)$ & check that it agrees with your *DFA* with φ as input.

- (20) 5. (a) Find a *context-free grammar* G which generates the language $L_5 = \{a^k b^n: n \geq 3k + 2, k \geq 0\} \cup \{b^k c^n: 0 \leq n \leq 2k+4\}$.
 (b) Find *derivations* in G for each of the strings: (i) $a^1 b^6$ and (ii) $b^1 c^5$.

- (15) 6. Let A , B , and C be languages based on the *alphabet* $\{0,1\}$.
 (a) Is it always true that $A^*(B \cup C) \subseteq (A^* \cdot B) \cup (A^* \cdot C)$?
 (b) Is it always true that $A \cdot (B - C) \subseteq (A \cdot B) - (A \cdot C)$? (Justify your answers.)

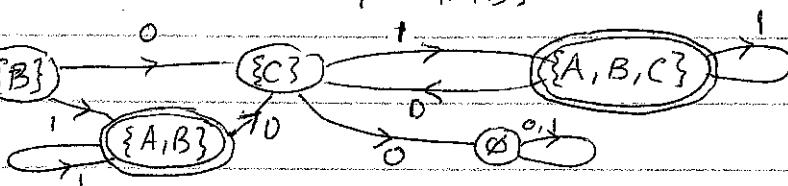
1 (a) A regular expression over $\{a, b, c, d\}$ is defined recursively as follows.

- (i) a, b, c, d, λ and \emptyset are regular expressions, and
 - (ii) if E & F are regular expr., then so are $(E+F)$, $(E.F)$, & E^* .

(b)

$$R(\lambda) = \{B\} \xrightarrow{\circ} \{A, B\} \xrightarrow{\circ} \{C\} \xrightarrow{\circ} \{A, B, C\} \xrightarrow{\circ} \{C\}$$

$$1.5 \text{ } M_{\odot} =$$



2(a) - ...bab...aab..., ...aab...bab..., ...aabab...

$$E_1 = (\underline{a} + \underline{b})^* \left(\underline{bab} \cdot (\underline{a} + \underline{b})^* \underline{aab} + \underline{aab} \cdot (\underline{a} + \underline{b})^* \underline{bab} + \underline{aabab} \cdot (\underline{a} + \underline{b})^* \right)$$

(b) ~~- - - 0 ... 10 ... 0 ... 10 (no 11's) or - - - 0 ... 10 ... 11 ... 0 ... (one 11)~~

$$E_2 = (\underline{0+10})^* + (\underline{0+10}).1 + (\underline{0+10})^*.11 \cdot (\underline{0+01})^*$$

3 (a) Two states p & q are indistinguishable in a DFA M if

for each $\varphi \in T^*$, $s^*(q, \varphi) \in A(M) \iff s^*(p, \varphi) \in A(M)$.

(b) $P_0 : \{A, D, E, G\} \quad \{B, C, F\}$ (non-acc. & acc. states)

$$P_1 : \{A, D\} \quad \{E, G\} \quad \{B, C, F\}$$

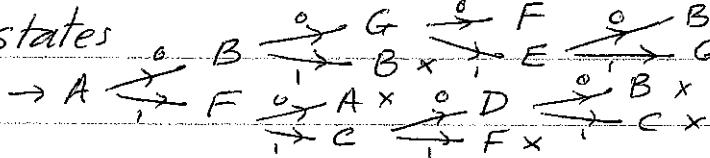
$$P_2 : \{A, D\} \quad \{E, G\} \quad \{B\} \quad \{C, F\}$$

$P_3 : \{A, A\} \{E, EG\} \{B\} \{C, F\}$

$P_{\text{E}} = \{A, A^2, E, E^2, B, B^2, C, F\}$

$$t \rightarrow F \rightarrow B$$

Check for inacc. states



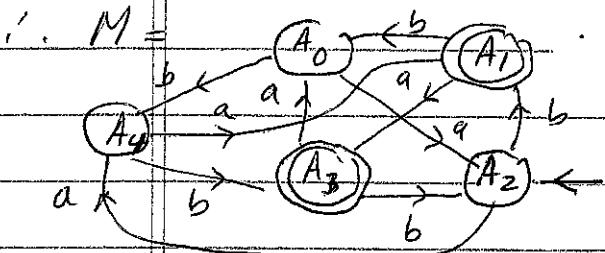
no inacc.
states.

| | | | | | |
|---|------------|---------|------------|---------|------------|
| | $\{A, D\}$ | $\{E\}$ | $\{G\}$ | $\{B\}$ | $\{C, F\}$ |
| 0 | $\{B\}$ | $\{B\}$ | $\{C, F\}$ | $\{G\}$ | $\{A, D\}$ |
| 1 | $\{C, F\}$ | $\{G\}$ | $\{E\}$ | $\{B\}$ | $\{C, F\}$ |

4(a) Let A_i ($i=0,1,2,3,4$) keep track of the output as each initial segment of the input is processed. Then A_1 & A_3 will be accepting states and since $f(\lambda) = 2n_a(\lambda) - n_b(\lambda) - 3 \equiv 2 \pmod{5}$, A_2 will be the initial st.

$$\text{Now } f(wa) = 2n_a(wa) - n_b(wa) - 3 \equiv [2n_a(w) - n_b(w) - 3] + 2 = f(w) + 2 \pmod{5}$$

$$\text{& } f(wb) = 2n_a(wb) - n_b(wb) - 3 \equiv [2n_a(w) - n_b(w) - 3] - 1 = f(w) - 1 = f(w) + 4 \pmod{5}$$



Check : $\xrightarrow{a} A_2 \xrightarrow{b} A_4 \xrightarrow{a} A_3 \xrightarrow{b} A_0 \xrightarrow{a} A_4 \xrightarrow{b} A_1 \text{ STOP}$

5(a) $\rightarrow S$, $S \rightarrow A$, $S \rightarrow B$ (gives the union)

$A \rightarrow aAbbbb|D$, $D \rightarrow Db$, $D \rightarrow bb$ gives $\{a^k b^n : k \geq 3k+2\}$

$B \rightarrow bBCC|CCCC$, $C \rightarrow c|\lambda$ gives $\{b^k c^n : 0 \leq n \leq 2k+4\}$

$$(b) (i) \rightarrow S \Rightarrow A \Rightarrow aAbbb \Rightarrow aDb^3 \Rightarrow abbb^3 = a'b^6$$

$$(ii) \rightarrow S \Rightarrow B \Rightarrow bBCC \Rightarrow bC^4CC \Rightarrow bC^4C\lambda \Rightarrow bC^4c \Rightarrow bC^3cc$$

$$\Rightarrow bC^2ccc \Rightarrow bCcccc \Rightarrow bcccccc = bC^5$$

6(a) Yes. Let $\varphi \in A^*(B \cup C)$. Then we can find $\alpha \in A^*$ & $\beta \in B \cup C$ such that $\varphi = \alpha \cdot \beta$. Since $\beta \in B \cup C$, $\beta \in B$ or $\beta \in C$. So

$\alpha \in A \& \beta \in B$ or $\alpha \in A \& \beta \in C$. $\therefore \varphi = \alpha \cdot \beta \in A^* \cdot B$ or $\varphi = \alpha \beta \in A^* \cdot C$
 $\therefore \varphi \in (A^* \cdot B) \cup (A^* \cdot C)$. Hence $A^* \cdot (B \cup C) \subseteq (A^* \cdot B) \cup (A^* \cdot C)$

(b) No. [Let us first try to prove $A \cdot (B - C) \subseteq (A \cdot B) - (A \cdot C)$. Let

$\varphi \in A \cdot (B - C)$. Then we can find $\alpha \in A$ & $\beta \in B - C$ so that $\varphi = \alpha \cdot \beta$.

Since $\beta \in B - C$, $\beta \in B$ and $\beta \notin C$. So $\varphi = \alpha \cdot \beta \in A \cdot B$. But we cannot say that $\varphi \notin A \cdot C$ because there could be $\alpha' \in A$ &

$\beta' \in C$ such that $\alpha' \cdot \beta' = \varphi$ and $\varphi \in A \cdot C$. Of course α' cannot

be α . Let $A = \{0, 01\}$, $B = \{1\}$, and $C = \{1\}$. Then

$$(A \cdot B) \cdot (B \cdot C) = S_0 \cdot S_1 S_2 S_3 \cdot S_0 S_1 S_2 S_3 = S_0 S_1 S_2 S_3 = \{011\}. \text{ Also}$$

$\therefore A.(B-C) \notin (A.B) - (A.C)$ in this particular case & so result is not always true.