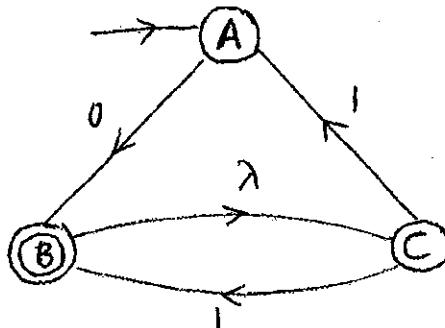


Answer all 6 questions. Provide all reasoning and show all working. An unjustified answer will receive little or no credit.
BEGIN EACH QUESTION ON A SEPARATE PAGE.

- (15) 1. (a) Define what is a **regular expression** over the alphabet $\{0,1,2\}$.

- (b) Convert the NFA on the right into an equivalent DFA.



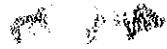
- (15) 2. Find regular expressions which describe the languages below.

- (a) $L_1 = \{\alpha \in \{a,b\}^*: \alpha \text{ contains both } ba \text{ & } abb \text{ as substrings}\}$
 (b) $L_2 = \{\beta \in \{a,b\}^*: \beta \text{ has at most two occurrences of } ab\}$

- (20) 3. (a) Find all the inaccessible states in the DFA below.
 (b) Then partition the remaining states into blocks of indistinguishable states and find the reduced machine.

	A	B	C	→ D	E	F	G	H
0	D	B	D	A	G	H	E	F
1	C	G	B	C	B	G	A	A

- (15) 4. Find a DFA which accepts precisely the strings in the language $L_4 = \{\omega \in \{a,b\}^*: f(\omega) > 1\}$, where $f(\omega) = [2n_b(\omega) - n_a(\omega) - 3] \pmod 4$, and then check your DFA with aaba as input.



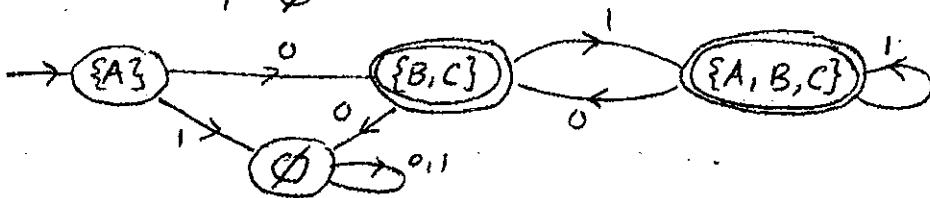
- (20) 5. (a) Define what is an **inherently ambiguous context free language**.
 (b) Find a context-free grammar which generates the language $L_5 = \{a^k b^n: k > 3n\} \cup \{b^k c^{n+2}: k < 2n+1\}$.

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

1(a) A regular expression over $\{0,1,2\}$ is defined recursively as follows. (i) $\lambda, 0, 1, 2$ and \emptyset are regular expressions. (ii) If E & F are reg. expr., then so are $(E+F)$, $(E \cdot F)$ & (E^*) .

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

DFA is



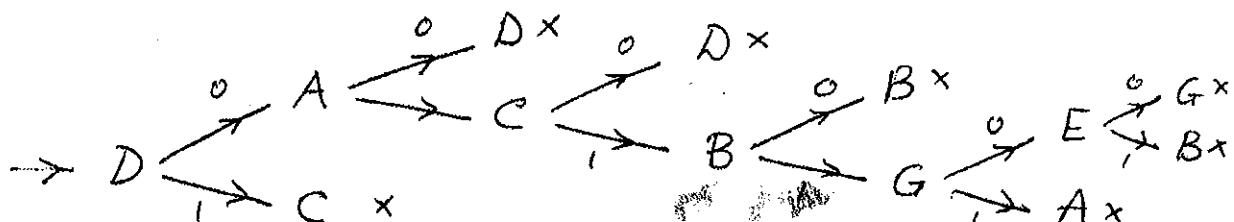
2(a) ...ba...abb..., ...abb...ba..., ...abba..., ...bab...b

$$E_1 = (\underline{a+b})^* \cdot (\underline{ba} \cdot (\underline{a+b})^* \cdot \underline{abb} + \underline{abb} \cdot (\underline{a+b})^* \cdot \underline{ba} + \underline{abb} \underline{a} + \underline{bab} \underline{b}) (\underline{a+b})^*$$

(b). No ab's , one ab only, two ab's only.

$$E_2 = \underline{b^*a^*} + \underline{b^*a^*ab} \cdot \underline{b^*a^*} + \underline{b^*a^*ab} \cdot \underline{b^*a^*ab} \cdot \underline{b^*a^*}$$

3(a)



i. F and H are inaccessible states

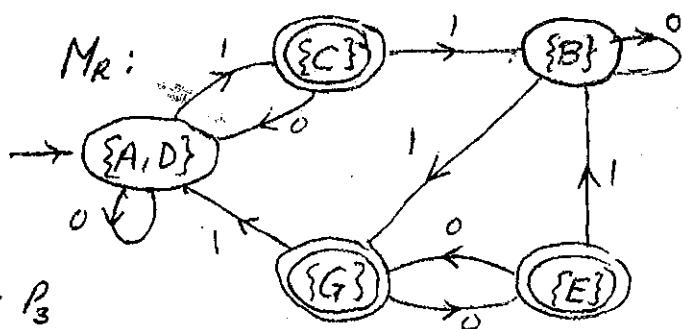
(b). P_0 : $\{A, B, D\}$ $\{C, E, G\}$

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

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

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

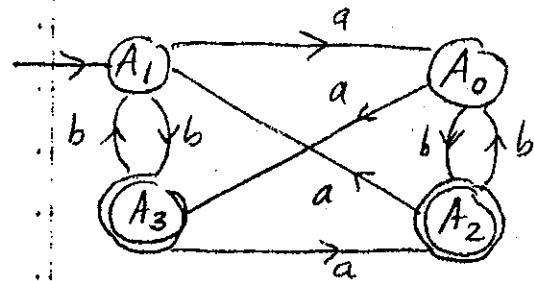
$$P_4 : \{A, D\} \{B\} \{C\} \{E\} \{G\} = P_3$$



4(a) Let A_i ($i=0,1,2,3$) be the state that stores the information that $f(w) \equiv i \pmod{4}$. $f(\lambda) = 2n_b(\lambda) - n_a(\lambda) - 3 = -3 \equiv 1 \pmod{4}$, so A_1 will be the initial state. Also A_2 & A_3 will be the accepting states since $f(w) > 1$ when $f(w) = 2$ or $3 \pmod{4}$. Finally

$$f(wa) = 2n_b(wa) - n_a(wa) - 3 = 2n_b(w) - n_a(w) - 3 - 1 = f(w) + 3 \pmod{4}$$

$$f(wb) = 2n_b(wb) - n_a(wb) - 3 = 2n_b(w) - n_a(w) - 3 + 2 = f(w) + 2 \pmod{4}$$



(b) Input: $a \ a \ b \ a$
states: A_1, A_0, A_3, A_1, A_0

$$\text{check: } f(aaba) = 2(1) - 3 - 3 \\ = -4 \equiv 0 \pmod{4} \checkmark$$

5(a) An inherently ambiguous CFL is a language which can be generated by a CFG but cannot be generated by an unambiguous CFG.

(b) $S \rightarrow A/B$ gives the union
 $A \rightarrow aaaAb/aA/a$ gives $\{a^{3n+p}, b^n : n \geq 0, p \geq 0\}$
 $B \rightarrow DDBc/cC, D \rightarrow b/\lambda$ gives $\{b^k, c^{n+2} : k \leq 2n\}$
 $S \Rightarrow A \Rightarrow a^3Ab \Rightarrow a^6Ab^2 \Rightarrow \dots \Rightarrow a^{3n}Ab^n \Rightarrow \dots \Rightarrow a^{3n+p}Ab^n \Rightarrow a^{3n+p+1}b^n$
 $S \Rightarrow B \Rightarrow D^2Bc \Rightarrow D^4Bc^2 \Rightarrow \dots \Rightarrow D^{2n}Bc^n \Rightarrow D^{2n}cc.c^n \Rightarrow b^k.c^{2n+2}$

6(a) YES. Let $\varphi \in (A.D^c) \cup (B.D^c)$. Then $\varphi \in A.D^c$ or $\varphi \in B.D^c$. In the first case $\varphi = \alpha.\gamma$ with $\alpha \in A$ & $\gamma \in D^c$, so $\varphi \in (A \cup B).D^c$. And in the second, $\varphi = \beta.\gamma$ with $\beta \in B$ and $\gamma \in D^c$, so $\varphi \in (A \cup B).D^c$ again. So in either case $\varphi \in (A \cup B).D^c$. $\therefore (A.D^c) \cup (B.D^c) \subseteq (A \cup B).D^c$

(b) NO. Let $A = \{a\}$, $B = \{ab\}$, and $D = \{\lambda, b\}$. Then $A.D - B.D = \{a\}.\{\lambda, b\} - \{ab\}.\{\lambda, b\} = \{a, ab\} - \{ab, abb\} = \{a\}$ and $(A-B).D = (\{a\} - \{ab\}).\{\lambda, b\} = \{a\}.\{\lambda, b\} = \{a, ab\}$. So $ab \in (A-B).D$ but $ab \notin A.D - B.D$. Hence $(A-B).D \not\subseteq (A.D) - (B.D)$.