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.

2. Find regular expressions which describe the languages below.
   (a) \(L_1 = \{\alpha \in \{a,b\}^* : \alpha \text{ contains both } ab \text{ & } abb \text{ as substrings}\}\)
   (b) \(L_2 = \{\beta \in \{a,b\}^* : \beta \text{ has at most two occurrences of } ab\}\)

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.

4. Find a DFA which accepts precisely the strings in the language \(L_4 = \{\omega \in \{a,b\}^* : f(\omega) > 1\}\), where \(f(\omega) = [2n_3(\omega) - n_4(\omega) - 3] \mod 4\), and then check your DFA with aaba as input.

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

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 \((A-B) \cdot D \subseteq (A-D)-(B \cdot D)\)?
      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.F) \& (E*)

(b) \[
R(\lambda) = \{A\} \quad R(0) = \{B, C\} \quad R(1) = \{A, B, C\} \quad R(2) = \{B, C\}\]

DFA is

---

2(a) \ldots ba... abb..., ...abb...ba..., ...abba..., ...bab...
\[E_1 = (a+b)^* (ba.(a+b)^* abb + abb.(a+b)^* ba + abba + babb)(a+b)^*\]

(b) No ab's, one ab only, two ab's only.
\[E_2 = b^*a^* + b^*a^*ab + b^*ab + b^*a^*ab + b^*a^*\]

3(a)

---

(b) \[P_0: \{A, B, D\} \{C, E, G\}\]
\[P_1: \{A, B, D\} \{C\} \{E, G\}\]
\[P_2: \{A, D\} \{B\} \{C\} \{E, G\}\]
\[P_3: \{A, D\} \{B\} \{C\} \{E\} \{G\}\]
\[P_4: \{A, D\} \{B\} \{C\} \{E\} \{G\} = P_3\]
4(a) Let $A_i$ ($i=0,1,2,3$) be the state that stores the information that $f(w) \equiv i \pmod{4}$. $f(\lambda) = 2n_b(\lambda) - n_a(\lambda) - 3 \equiv 1 \pmod{4}$, 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 $f(\omega a) = 2n_b(\omega a) - n_a(\omega a) - 3 = 2n_b(\omega) - n_a(\omega) - 3 - 1 = f(\omega) + 3 \pmod{4}$ $f(\omega b) = 2n_b(\omega b) - n_a(\omega b) - 3 = 2n_b(\omega) - n_a(\omega) - 3 + 2 = f(\omega) + 2 \pmod{4}$

(b) Input: $a$ $a$ $a$ $b$ $a$

States: $A_1$, $A_0$, $A_3$, $A_1$, $A_0$

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

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) $S \rightarrow A/B$

$A \rightarrow aaaaAb/aaA/a$

$\quad \quad \quad \quad$ gives $\{a^{3n+1+p}.b^n : n \geq 0, p \geq 0\}$

$B \rightarrow BBc/Cc, D \rightarrow b/\lambda$

$\quad \quad \quad \quad$ gives $\{b^k.c^{n+2} : k \leq 2n\}$

$S \Rightarrow A \Rightarrow a^3Ab \Rightarrow a^6Ab^2 \Rightarrow \ldots \Rightarrow a^{3n}Ab^n \Rightarrow \ldots \Rightarrow a^{3n+1+2p}.b^n$

$S \Rightarrow B \Rightarrow D^2Bc \Rightarrow D^4Bc^2 \Rightarrow \ldots \Rightarrow D^{2n}Bc^n \Rightarrow D^{2n}Cc^n \Rightarrow b^k.c^{2n+2}$

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 $\varphi = \alpha \beta$ with $\alpha \in A \& \beta \in D^c$, so $\varphi \in (A \cup B).D^c$. And in the second, $\varphi = \beta \gamma$ with $\beta \in B$ and $\gamma \in D$, so $\varphi \in (A \cup B).D^c$ again. So in either case $\varphi \in (A \cup B).D^c$.

(b) NO. Let $A = \{a\}$, $B = \{ab\}$, and $D = \{\lambda, b\}$. Then $A.D = \{a\}.\{\lambda, b\} - \{ab\}$. $\{\lambda, b\} = \{a, ab\} - \{ab, abb\} = \{a\}$

and $(A-B).D = \{a\} - \{ab\}$. $\{\lambda, b\} = \{a\} \& \{\lambda, b\} = \{a, ab\}$. So $a \in (A-B).D$ but $a \notin A.D - B.D$. Hence $(A-B).D \notin (A.D)-(B.D)$. 