Chapter 6

Problem 1:
For the regular expressions:
\((a \cup bc \cup c)^*\) in our posted solutions to Exercise 25 of Chapter 2 in Homework 1
\((b^*ab^*ab^*)^* \cup b^*\) in our posted solutions to Exercise 26 of Chapter 2 in Homework 1.

1. Construct a finite automaton.
2. Convert your finite automaton into an equivalent regular grammar.

Solution 1:

For regular expression: \((a \cup bc \cup c)^*\)

part 1

Figure 1: Basic NFAs for \(a\), \(b\), \(c\) and \(bc\)
By combining the NFAs above with $\lambda$-transitions we get the NFA below:

![Image of NFA](image)

Figure 2: Combining NFAs to create NFA with $\lambda$-transitions

A reduced NFA can be obtained by the following steps:

- $q_0, q_1, q_2, q_7,$ and $q_4$ are merged into one state called $Q_0$
- $q_8$ and $q_9$ are merged into one state called $Q_1$
- $q_3, q_5, q_6, q_{10}$, and $q_{11}$ into one state called $Q_2$

![Image of Reduced NFA](image)

Figure 3: Reduced NFA

**part 2**

Based on the NFA in Figure 3, we construct the grammar:

- $S \rightarrow aQ_2 | cQ_2 | bQ_1 | Q_2$
- $Q_1 \rightarrow cQ_2$
- $Q_2 \rightarrow S | \lambda$

On removing the chain rules we get:

- $S \rightarrow aQ_2 | cQ_2 | bQ_1 | \lambda$
- $Q_1 \rightarrow cQ_2$
- $Q_2 \rightarrow aQ_2 | cQ_2 | bQ_1 | \lambda$
For regular expression: \( (b^*ab^*ab^*)^* \cup b^* \)

part 1

![NFA Diagram](image)

Figure 4: NFA for Chap2 26

part 2

Based on the NFA in Figure 4, we construct the grammar:

\[
S \rightarrow Q_1 \mid Q_5 \mid Q_6 \\
Q_1 \rightarrow bQ_1 \mid aQ_2 \\
Q_2 \rightarrow bQ_2 \mid aQ_3 \\
Q_3 \rightarrow bQ_3 \mid aQ_4 \\
Q_4 \rightarrow bQ_4 \mid Q_1 \mid Q_6 \\
Q_5 \rightarrow bQ_5 \mid Q_6 \\
Q_6 \rightarrow \lambda
\]

Removing \text{chain rules} we obtain the following grammar which is in regular form.

\[
S \rightarrow bQ_1 \mid aQ_2 \mid \lambda \\
Q_1 \rightarrow bQ_1 \mid aQ_2 \\
Q_2 \rightarrow bQ_2 \mid aQ_3 \\
Q_3 \rightarrow bQ_3 \mid aQ_4 \\
Q_4 \rightarrow bQ_4 \mid bQ_1 \mid aQ_2 \mid \lambda \\
Q_5 \rightarrow bQ_5 \mid \lambda \\
Q_6 \rightarrow \lambda
\]

Problem 2: For the NFAs from:
Exercise 23 of Chapter 5 and
Exercise 36 of Chapter 5

1. Convert the finite automaton into an equivalent regular expression.
2. Convert your finite automaton into an equivalent regular grammar.

Solution 2:
Exercise 23 of Chapter 5

part a
Figure 5: NFA for Chap5 Question 23

Figure 6: Step1: Remove $q_1$ from NFA Chap 5-23

Start to eliminate the state $q_1$, and the result is shown in Figure 6. The regular expression is

$$a^*(ab^+)(ab^+ \cup aa^*ab^+)^*$$

Note: this regular expression is equivalent to $(a^+b^+)^+$. 

$$
\begin{align*}
& a^*(ab^+)(ab^+ \cup aa^*ab^+)^* \\
& \equiv a^+b^+(ab^+ \cup aa^*b^+)^* \\
& \equiv a^+b^+(a^+b^+)^* \\
& \equiv (a^+b^+)^+
\end{align*}
$$

**part b**

Based on the NFA in Figure 5, we can construct the regular grammar:

$$
\begin{align*}
S & \rightarrow aS \mid aQ_1 \\
Q_1 & \rightarrow bQ_1 \mid bQ_2 \\
Q_2 & \rightarrow aQ_1 \mid aS \mid \lambda
\end{align*}
$$

Exercise 36 of Chapter 5

**part a**
Figure 7: NFA for Chap5 Question 36

Figure 8: Step 1: Create a new accepting state

Figure 9: Step 2: remove State $q_1$

Figure 10: Step 3: remove State $q_2$

Figure 11: Step 3: removing the loop on $q_0$
The regular expression is: \( a^* b^+ c^* \cup a^* b^* \cup a^* c^+ \)

**part b**

Based on the NFA in Figure 7, we can construct the grammar:

\[
S \rightarrow aS \mid cQ_1 \mid Q_2 \\
Q_1 \rightarrow cQ_1 \mid \lambda \\
Q_2 \rightarrow bQ_2 \mid bQ_1 \mid \lambda
\]

On removing the chain rule we get the regular grammar:

\[
S \rightarrow aS \mid cQ_1 \mid bQ_2 \mid bQ_1 \mid \lambda \\
Q_1 \rightarrow cQ_1 \mid \lambda \\
Q_2 \rightarrow bQ_2 \mid bQ_1 \mid \lambda
\]

**Problem 3:**

For the regular grammar in our posted solutions of Exercise 9 of Chapter 4 in Homework 2, and the regular grammar for solution of Exercise 25 of Chapter 3 in Homework 1.

1. Construct a finite automaton based on the grammar.
2. Convert your finite automaton into an equivalent regular expression.

**Solution 3:**

Regular grammar in our posted solutions of Exercise 9 of Chapter 4 in Homework 2:

\[
S \rightarrow aA \mid a \mid cC \mid c \mid bB \mid b \\
A \rightarrow aA \mid a \mid bB \mid b \\
B \rightarrow bB \mid b \\
C \rightarrow cC \mid c \mid bB \mid b
\]

**part a**

```

part b
We eliminate the state A first, and then state C, and state B to get the regular expression. The detailed steps are shown in Figure 12 to 20.
```

6
Figure 12: Step 1: remove State A

\[ a^* \overline{a} U a^* b U a \overline{b} U c = a^* U a^* b U U c \]

\[ a^* b = a^* b \]

\[ b \]

\[ b \]

\[ b \]

\[ b \]

\[ a^* U b U U c \]

Figure 13: Step 2: remove State C

\[ a^* U a^* b U U U c \]

\[ a^* b \]

\[ b \]

\[ b \]

\[ b \]

\[ cc^*(b U c) = c^* b U cc^* \]

Figure 14: Step 3: remove State B

\[ a^* U a^* b U U U c U c^* b U cc^* = a^* U c^* U a^* b U U U c^* b \]

\[ (a^* b U U U c^* b) b^* = (a^* b U U U c^* b) b^* \]

\[ c^* b U cc^* \]

\[ (a^* b U b U c^* b) b^* = a^* b^* U b b^* U c b b^* \]

Figure 15: Step 3: reduce the number of arcs
We can see the regular expression is
\[ a^+ \cup c^+ \cup a^+b \cup b \cup c^+b \cup a^+bb^+ \cup bb^+ \cup c^+bb^+ \]

Actually, it is equivalent to the regular expression \( a^+ \cup c^+ \cup (a^+b \cup b \cup c^+)b^* \) in our HW2 solution, because:
\[
\begin{align*}
\equiv & \quad a^+ \cup c^+ \cup a^+b \cup b \cup c^+b \cup a^+bb^+ \cup bb^+ \cup c^+bb^+ \\
\equiv & \quad a^+ \cup c^+ \cup b^+ \cup a^+b \cup c^+b \cup a^+bb^+ \cup c^+bb^+ \\
\equiv & \quad a^+ \cup c^+ \cup a^+bb^* \cup bb^* \cup c^+bb^* \\
\equiv & \quad a^+ \cup c^+ \cup (a^+b \cup b \cup c^+)b^*
\end{align*}
\]

Regular grammar for solution of Exercise 25 of Chapter 3 in Homework 1
\[
\begin{align*}
S & \to aA \mid bC \mid aB \mid bD \mid \lambda \\
C & \to aA \mid bC \mid \lambda \\
A & \to aC \mid bA \\
D & \to aD \mid bB \mid \lambda \\
B & \to aB \mid bD
\end{align*}
\]

part a

Figure 16: NFA obtained from the grammar
Figure 17: Step 1: Adding a new accepting state

Figure 18: Step 2: remove State C
Figure 19: Step 3: remove State $D$

Figure 20: Step 4: remove State $A$

Figure 21: Step 5: remove State $B$
The regular expression is:
\[b^+ \cup \lambda \cup ba^* \cup b^+ab^* \cup ab^*ab^* \cup aa^*ba^* \cup ba^*ba^*
\equiv b^+ \cup \lambda \cup ba^* \cup b^+ab^* \cup ab^*ab^* \cup a^*ba^* \cup ba^*ba^*\]

**Problem 4: Solution 4:**

**Chap 6.7.a**

Let \( H = \{ w \in L \text{ and } w \text{ ends with } aa \} \)

Let \( L_1 \) be the language over \( \{ a, b, c \} \) that contains strings ending with \( aa \). \( L_1 \) is described by the regular expression \((a \cup b \cup c)^*aa \). And so \( L_1 \) is regular.

A language that contains all strings that belong to both \( L \) and \( L_1 \) can be obtained by the intersection of the two languages. Therefore \( H = L \cap L_1 \). The regularity of \( H \) then follows from the closure of the regular languages under intersection.

**Chap 6.7.b**

Let \( H = \{ w \in L \text{ or } w \text{ contains an } a \} \)

Let \( L_1 \) be the language over \( \{ a, b, c \} \) of strings that contain an \( a \). \( L_1 \) is described by the regular expression \((a \cup b \cup c)^*a(a \cup b \cup c)^* \). And so \( L_1 \) is regular.

A language that contains any string that belongs to either \( L \) or \( L_1 \) or both, can be obtained by the union of the two languages. Therefore \( H = L \cup L_1 \). The regularity of \( H \) then follows from the closure of the regular languages under union.

**Chap 6.7.c**

Let \( H = \{ w \in L \text{ is a palindrome over } \{ a, b \} \} \)

Any \( w \notin L \) belongs to \( \bar{L} \). We know that \( \bar{L} \) is regular as regular languages are closed under complement.

Let \( L_1 \) be the language over \( \{ a, b, c \} \) of strings that contain an \( a \). We have shown in the previous part(b) that this language is regular. Any \( w \) that does not contain an \( a \) then belongs to \( \bar{L}_1 \). We know that \( \bar{L}_1 \) is regular as regular languages are closed under complement.

A language that contains all strings that belong to both \( \bar{L} \) AND \( \bar{L}_1 \), can be obtained by the intersection of the two languages. Therefore \( H = \bar{L} \cap \bar{L}_1 \). The regularity of \( H \) then follows from the closure of the regular languages under complement and intersection.

**Chap 6.7.d**

Let \( H = \{ uv | u \in L \text{ and } v \notin L \} \)

Any \( v \notin L \) belongs to \( \bar{L} \). We know that \( \bar{L} \) is regular as regular languages are closed under complement.

A language that contains strings formed by the concatenation of two strings belonging to two separate languages, can be obtained by the concatenation of the two languages. Therefore \( H = \bar{L}L \). The regularity of \( H \) then follows from the closure of the regular languages under complement and concatenation.

**Chap 6.14.a**

By way of contradiction, we assume \( L = \{ w \mid w \text{ is a palindrome over } \{ a, b \} \} \) is regular. Let \( M \) be a DFA that accepts \( L \), and \( k \) be the number of states in \( M \). Consider the string \( z \) equal to \( a^kba^k \). Clearly, \( z \in L \).
By the pumping lemma, $z$ can be written as $uvw$ where:

1. $v \neq \lambda$
2. $\text{length}(uv) \leq k$
3. $uv^i w \in L$ for all $i \geq 0$

However, by condition 2, $v$ must consist of only $a$’s. Pumping $v$ would produce the string $uv^2 w$ where the number of $a$ before the $b$ is more than the number of $a$ after the $b$. Therefore, $uv^2 w$ is not a palindrome, and $uv^2 w \notin L$, yielding a contradiction.

Thus, $L$ is not regular.

Chap 6. 14. b
By way of contradiction, we assume $L = \{a^n b^m | n < m\}$ is regular. Let $M$ be a DFA that accepts $L$, and $k$ be the number of states in $M$. Consider the string $z$ equal to $a^k b^{k+1}$. Clearly, $z \in L$.

By the pumping lemma, $z$ can be written as $uvw$ where:

1. $v \neq \lambda$
2. $\text{length}(uv) \leq k$
3. $uv^i w \in L$ for all $i \geq 0$

However, by condition 2, $v$ must consist of only $a$’s. Pumping $v$ would produce the string $uv^2 w$ which contains at least as many $a$’s and $b$’s. Therefore, $uv^2 w \notin L$, yielding a contradiction. Thus, $L$ is not regular.

Chap 6. 14. c
By way of contradiction, we assume $L = \{a^i b^j c^{2j} | i \geq 0, j \geq 0\}$ is regular. Let $M$ be a DFA that accepts $L$, and $k$ be the number of states in $M$. Consider the string $z$ equal to $b^k c^{2k}$. Clearly, $z \in L$.

By the pumping lemma, $z$ can be written as $uvw$ where:

1. $v \neq \lambda$
2. $\text{length}(uv) \leq k$
3. $uv^i w \in L$ for all $i \geq 0$

However, by condition 2, $v$ must consist of only $b$’s. Pumping $v$ would produce the string $uv^2 w$ which could not contain as twice as many $c$’s as $b$’s. Therefore, $uv^2 w \notin L$, yielding a contradiction. Thus, $L$ is not regular.

Chap 6. 14. d
By way of contradiction, we assume $L = \{ww | w \in \{a, b\}^*\}$ is regular. Let $M$ be a DFA that accepts $L$, and $k$ be the number of states in $M$. Consider the string $z$ equal to $a^k b a^k b$. Clearly, $z \in L$.

By the pumping lemma, $z$ can be written as $uvw$ where:

1. $v \neq \lambda$
2. $\text{length}(uv) \leq k$
3. $uv^i w \in L$ for all $i \geq 0$
However, by condition 2, \(v\) must consist of only \(a\)'s. Pumping \(v\) would produce the string \(uv^2w\) where the number of \(a\)'s before the first \(b\) is greater than the number of \(a\)'s between the two \(b\)s. Therefore, \(uv^2w \notin L\), yielding a contradiction. Thus, \(L\) is not regular.

**Chap 6. 14. f**

\(L\) is the set of string over \(\{a, b\}^*\) in which the number of \(a\)'s is a perfect cube. By way of contradiction, we assume \(L\) is regular. Let \(M\) be a DFA that accepts \(L\), and \(k\) be the number of states in \(M\). Consider the string \(z\) equal to \(a^{k^3}\). Clearly, \(z \in L\), because \(\text{number of } a(z) = k^3\).

By the pumping lemma, \(z\) can be written as \(uvw\) where:

1. \(v \neq \lambda\)
2. \(\text{length}(uv) \leq k\)
3. \(uv^iw \in L\) for all \(i \geq 0\)

However, by condition 1, \(v\) must not be \(\lambda\). It means that \(0 < \text{length}(v) \leq k\). Because \(v\) consists of \(a\)s, we have \(\text{number of } a(v) = \text{length}(v)\), and \(0 < \text{number of } a(v) \leq k\). This observation can be used to compute the upper bound of \(\text{number of } a(uv^2w)\):

\[
\text{number of } a(uv^2w) = \text{number of } a(uvw) + \text{number of } a(v) \\
= k^3 + \text{length}(v) \\
\leq k^3 + k \\
< k^3 + 3k^2 + 3k + 1 \\
= (k + 1)^3
\]

Thus, \(uv^2w\) must not be in \(L\). The assumption that \(L\) is regular yields a contradiction and therefore \(L\) is not regular.

**Chap 6. 15**

Prove that the set of nonpalindromes over \(\{a, b\}\) is not a regular language.

We shall prove this by way of contradiction. Let us assume that \(H\) be the set of nonpalindromes over \(\{a, b\}\) and that \(H\) is regular. Then \(\overline{H}\) that is the set of palindromes over \(\{a, b\}\) will also be regular. However we have proved in Exercise 6.14, part(a) that \(\overline{H}\) is not regular. This implies that the complement of \(\overline{H}\) that is equal to \(H\) is also not regular. This contradicts our assumption of \(H\) being regular.

**Chap 6. 16**

Let \(L\) be a regular language and let \(L_1 = \{uu | u \in L\}\) be the language \(L\) “doubled”. Is \(L_1\) necessarily regular? Prove your answer.

No, \(L_1\) is not necessarily regular. Let us take the language \(\{ww | w \in \{a, b\}^*\}\) in exercise 6.14, part(d). We have shown that this language is not regular even though the language \(\{a, b\}^*\) is regular.