Problem 1

Give short answers to each of the following. Be sure to adequately explain your answers for full credit.

a. Give an example of a regular language \( R \) and a non-regular language \( N \), such that \( R \cap N \) is a regular language.

\[
R = ab^* \\
N = \{a^n b^n \mid n \geq 0\} \\
R \cap N = \{ab\}
\]

b. True or false: If an NFA with \( n \) states accepts no string with length less than \( n \), then it must accept no strings at all.

True. Suppose that the NFA accepts some string. Then there is a path from the start state to a final state. This implies that there is a simple path (in which every state appears at most once) from the start state to the final state. (A simple path is obtained by removing loops.) A simple path can contain at most \( n \) nodes and therefore at most \( n - 1 \) transitions. Therefore, a simple path defines an accepted string with length at most \( n - 1 \). If no such string exists in \( L(M) \), then the NFA accepts no strings at all, since it must be possible to take at least one simple path to a final state.

c. Consider this (false) claim: “If \( L_1 \) is a regular language and \( L_1 \cup L_2 \) is also regular, then \( L_2 \) must be regular”.

True or false: If the claim were true for all \( L_1 \) and \( L_2 \), then all languages would be regular.

True. Let \( L_1 = \Sigma^* \), which is clearly regular because it is denoted by the regular expression \( \Sigma^* \). Let \( L_2 \) be any language. By the definition of a language, \( L_2 \subseteq L_1 \). Therefore, \( L_1 \cup L_2 = L_1 \) and, hence, is regular. Since \( L_2 \) is any arbitrary language, we would have that all languages are regular.

The statement itself is clearly false though. Let \( L_1 = \Sigma^* \), \( L_2 = \{a^n b^n \mid n \geq 0\} \). Then \( L_1 \) is regular, and \( L_1 \cup L_2 = L_1 \) is regular, but we know \( L_2 \) is not regular, disproving the statement.
Problem 2

Let $M$ be the NFA-$\varepsilon$ below. The input alphabet is $\{a, b, c\}$.

Using the method described in the book and in class, construct a deterministic finite automaton that is equivalent to $M$. Show your steps.

First compute the $\varepsilon$-closure for each state of the NFA:

$E(\{p\}) = \{p\}$
$E(\{q\}) = \{p, q\}$
$E(\{r\}) = \{p, q, r\}$

Then compute the transition function $\delta$:

$\delta(\{p\}, a) = E(\{p\}) = \{p\}$
$\delta(\{p\}, b) = E(\{q\}) = \{p, q\}$
$\delta(\{p\}, c) = E(\{r\}) = \{p, q, r\}$
$\delta(\{p, q\}, a) = E(\{p, q\}) = E(\{p\}) \cup E(\{q\}) = \{p, q\}$
$\delta(\{p, q\}, b) = E(\{q, r\}) = E(\{q\}) \cup E(\{r\}) = \{p, q, r\}$
$\delta(\{p, q\}, c) = E(\{r\}) = \{p, q, r\}$
$\delta(\{p, q, r\}, a) = E(\{p, q, r\}) = E(\{p\}) \cup E(\{q\}) \cup E(\{r\}) = \{p, q, r\}$
$\delta(\{p, q, r\}, b) = E(\{q, r\}) = \{p, q, r\}$
$\delta(\{p, q, r\}, c) = E(\{p, r\}) = \{p, q, r\}$
This gives us the DFA:
Problem 3

Convert the following NFA-ε into an equivalent regular expression using the technique shown in class and on pages 72–76 of our book.

Step 1 (add new start and final states)

Step 2 (after \(q_0\) removed)

Step 3 (after \(q_1\) removed)

Step 4 (after \(q_2\) removed) = Resulting regular expression
Problem 4

Using the construction method in the proof of Kleene’s theorem in the textbook, construct a nondeterministic finite automaton that accepts the language \(((a \cup b)(a \cup b)^{*}a)^{*}\).

Note that this is a different construction for Kleene star than is used by Sipser. Either way is fine.
NFA for \((a \cup b)(a \cup b)^* a\)

NFA for \(((a \cup b)(a \cup b)^*)^*\)
Problem 5

For each language, prove whether it is a regular language. If you say it is regular, give a DFA, NFA, or regular expression for \( L \). If you say it is not regular, prove this using the Pumping Lemma for regular languages and/or closure properties.

a. \( L = \{a^m b^n a^m b^n \mid m, n \geq 0\} \).

Non-regular.

We can use the Pumping Lemma to prove \( L \) is not regular:

By contradiction; assume that \( L \) is regular. Let \( p \) be the length guaranteed by the Pumping Lemma. Consider the string \( w = a^p b^p a^p b^p \). Then \( |w| = 4p \geq p \) and \( w \in L \). Therefore, there exist strings \( x, y, \) and \( z \) such that \( w = xyz \), \(|xy| \leq p\), and \( y \neq \varepsilon \), and for any non-negative integer \( i \), \( xy^i z \in L \).

Since \(|xy| \leq p\), \( y \) must consist solely of \( a \)'s. But then \( xy^2 z = a^{p+|y|} b^p a^p b^p \), and since \(|y| > 0\), we know \( xy^2 z \) will have at least one more \( a \) at the beginning of the string than there are \( b \)'s at the end. Therefore, \( xy^2 z \notin L \).

We have reached a contradiction, so our assumption was wrong, and \( L \) is not regular. \( \blacksquare \)

Or we can use closure properties to transform \( L \) into a known non-regular language:

\[ L \cap a^* b^* = \{a^i b^j \mid i \geq 0\} \]

\( a^* b^* \) is a regular language because it’s written as a regular expression.

Taking the intersection with a regular language preserves regularity, so if \( L \) is regular, then \( \{a^i b^j \mid i \geq 0\} \) is regular, but we’ve proven using the Pumping Lemma that this is a non-regular language.

Therefore, \( L \) must not be a regular language.

b. \( L = \{a^n \mid n \text{ is a multiple of } 8\} \)

Regular.

A regular expression for \( L \) is:

\( (aaaaaa)^* \)

c. \( L = \{x=y+z \mid x, y, z \text{ are binary integers and } x \text{ is the sum of } y \text{ and } z\} \). The alphabet is \( \Sigma = \{0, 1, =, +\} \); an example string is \( 1011 = 101 + 110 \).
Non-regular.

We can prove it using the Pumping Lemma:

By contradiction; assume $L$ is regular. Let $n$ be the length guaranteed by the Pumping Lemma. Consider the string $w = 1^n=1^n+0^n$. Then $|w| = 3n + 2 \geq n$ and $w \in L$. Therefore, there exist strings $x, y, \text{ and } z$ such that $w = xyz$, $|xy| \leq n$, and $y \neq \varepsilon$, and for any non-negative integer $i$, $xy^iz \in L$.

Since $|xy| \leq n$, $y$ must consist solely of 1s. But then $xy^0z = 1^n-|y|=1^n+0^n$, and since $|y| > 0$, we know $xy^0z$ has more 1s on the right side of the equal sign than the left.

We have reached a contradiction, so our assumption was wrong, and $L$ is not regular. ■
Problem 6

Let \( \Sigma = \{a, b\} \). Give a regular expression for the set of all strings in \( \Sigma^* \) with exactly one occurrence of the substring aaa.

\[
(b \cup ab \cup aab) \cdot aaa(b \cup ba \cup baa)^*
\]

The trick is to make sure that more than two a's do not have the opportunity to ever get next to each other, with the exception of the central aaa.