Exercise 1: Regular Expressions (6 Points)

Consider the following regular expressions. What language do they recognize? Give two strings that are members of the the corresponding language and two strings which are not members – a total of four strings for each part. Assume for the first two parts that the alphabet \( \Sigma = \{a, b, c\} \).

(a) \( a^* b^* c^* \)

(b) \( ((a \cup c)^* b (a \cup c)^* b (a \cup c)^* b (a \cup c)^*)^* \)

Give a regular expression for each of the following languages.

(c) \( L_1 \) is the language, over alphabet \( \{a, b\} \), of all strings starting and ending with the same symbol from the alphabet \( \Sigma \).

(d) \( L_2 \) is the language, over alphabet \( \{0, 1\} \), of all alternating 0 and 1 strings of length at least 2.

Sample Solution

(a) \( L = \{ w \in \Sigma^* \mid \text{the letters are in alphabetical order or } w \text{ is } \epsilon \} \). For example, \( aabb, aaaaabbeccc \) are members, whereas \( acba, bbcaeb \) are not.

(b) \( L = \{ w \in \Sigma^* \mid \text{the number of } b \text{'s is a non zero multiple of } 3 \text{ or the number of } a \text{'s, } b \text{'s, and } c \text{'s is zero } \} \). For example, \( abbc, bbb \) are members, whereas \( abc, bb \) are not.

(c) \( a(a \cup b)^* a \cup b(a \cup b)^* b \cup a \cup b \)

(d) \( (01)^+ \cup (10)^+ \cup 0(10)^+ \cup 1(01)^+ \)

Exercise 2: The Pumping Lemma: Sufficiency or Necessity? (4 Points)

Consider the language \( L = \{ c^m a^n b^n \mid m, n \geq 0 \} \cup \{a, b\}^* \) over the alphabet \( \Sigma = \{a, b, c\} \).

(a) Describe in words (not using the pumping lemma), why \( L \) cannot be a regular language.

(b) Show that the property described in the Pumping Lemma is a necessary condition for regularity but not sufficient for regularity.

\( \text{Hint: Use } L \text{ as counter example, i.e., show that it can be 'pumped' (in the sense of the pumping lemma), but is still not regular.} \)
Sample Solution

(a) For recognizing that a word has the same number of a’s and b’s, a DFA would have to count the number of appearances of these characters, requiring at least one state for each appearance. But as the number of appearances can be arbitrary large, the automaton would need an infinite number of states.

(b) We show that $L$ has the properties described in the Pumping Lemma. Then we show that for a language, having these properties do not imply regularity.

As the pumping length we choose an arbitrary $p \geq 1$. Let $x$ be some word of length at least $p$. We must show that there is a composition $x = uvw$ having the three properties from the lemma:

1. $|v| \geq 1$
2. $|uv| \leq p$
3. For all $i = 0, 1, 2, \ldots$ it holds: $uv^iw \in L$

This is clear if $x \in \{a, b\}^*$. So assume $x = c^ma^n b^n$ with $m \geq 1$. We can choose $u = \epsilon$, $v = c$ and $w = c^{m-1}a^n b^n$ as a composition of $x$ having properties 1, 2, 3.

Exercise 3: To Be Regular or Not to Be

Let $\Sigma = \{0, 1\}$, prove the following:

(a) The language $A = \{0^k w 0^k | k \geq 1 \text{ and } w \in \Sigma^* \}$ is regular.

(b) The language $B = \{0^k 1 w 0^k | k \geq 1 \text{ and } w \in \Sigma^* \}$ is not regular.

Sample Solution

(a) The language $A$ contains any string of length at least 2 which starts and ends with 0. This because the substring $w$ in the middle could be an arbitrary string, e.g., $0^k w 0^k = 0w0 \in A$, since $w' = 0^{k-1} w 0^{k-1} \in \Sigma^*$. Hence, we can easily get an automaton which accepts the language containing strings which start and end with 0.

(b) Intuitively, an automaton needs to count the number of 0s before 1, so that it can verify the given string contains at least equal number of 0s on the other end. However, we know that the finite automata can not count or memorize a large number, if $k$ is very large. Therefore, there would always exists a string (possibly very large) in $B$ which can not be accepted by a finite automaton. Thus $B$ can not be a regular language.

Formally, one can show a contradiction directly from the Pumping Lemma. Consider a string $s = 0^p 10^p$ (assuming $p > 0$ be the pumping length) that is in $B$. Let $s = xyz$ be a decomposition of the word that satisfies the requirements of the pumping lemma. As $|xy| \leq p$ we have that $x = 0^{p-r-q}, y = 0^q, z = 0^r 10^p$, for some $q \geq 1$ and $r \geq 0$. Then all the conditions of the Pumping Lemma are satisfied. Therefore, if $B$ is regular, then $xy^2z$ must be in $B$ (by Pumping Lemma). However, $xy^2z = 0^p 10^p$ does not belongs to $B$, which contradicts that $B$ is regular.

Exercise 4: NFAs to Regular Expressions

Consider the following NFA:
Give the regular expression defining the language recognized by this NFA by stepwise converting it into an equivalent GNFA with only two nodes.

**Sample Solution**

1) Add a new start and accepting state, connect them with $\epsilon$ transitions from/to the previous start/accept states, replace multiple labels with unions, add transitions with $\emptyset$ when not present in the original DFA (for a better readability, some edges with label $\emptyset$ are left out in the following diagram):

2) Rip off $q_1$:

3) Rip off $q_2$:
4) Rip off \( q_3 \):

\[
\epsilon (a \cup b) a^* b \\
\epsilon \cup a \\
(a(a \cup b) \cup b) a^* b
\]

\[
((a \cup b) a^* b) ((a(a \cup b) \cup b) a^* b) (\epsilon \cup a) \cup \epsilon
\]

\[
q_{\text{start}} \quad q_{\text{accept}}
\]