Problem 1

Transform the following regular expression into an equivalent NFA using the procedure studied in class.

\[ E = (a(ba)^*) \cup (b(a \cup b)^*) \]

Notice: Your automaton should be obtained using the method in the textbook. Giving an automaton for the right language, but not corresponding to the regular expression above, won’t receive full credit. You should give some (say at least 3) of the intermediate results of the transformation, and indicate the corresponding subexpression.

Problem 2

Transform the automaton shown in Figure 1 into a corresponding regular expression using the procedure studied in class. (This is the automaton corresponding to the language of all binary strings that encode a multiple of 3.)

![Figure 1: DFA for Problem 2](image)

Notice: as in Problem 1, you should use the procedure from the textbook. To facilitate grading, please remove the states in the order “A,B,C”, and show all intermediate GNFA's.

Problem 3

Give regular expressions corresponding to the following languages. All languages are over the alphabet \( \Sigma = \{0, 1\} \).

1. \( L_1 \): the set of all words that contain “00” as a substring.
2. $L_2$: the set of all words that do not contain “11” as a substring.
3. $L_3$: the set of all words of even length (i.e., length 0, 2, 4, etc.).
4. $L_4$: the set of all words whose length is exactly 3.

Problem 4

In the previous problem you proved that languages $L_1, \ldots, L_4$ are regular, by giving corresponding regular expressions. Prove that the following languages are also regular, using the regularity of $L_1, \ldots, L_4$ and using the closure properties of regular languages under union, concatenation, complement and star.

1. the set of all words that contain two consecutive 0’s, but no two consecutive 1’s.
2. the set of all words whose length is a multiple of 6.
3. The set of all words in which 0 and 1 always alternate, i.e., words of the form “010101...” or “10101...”. (This language includes the empty word $\epsilon$, and the singletons “0”, and “1”.
4. The set of all words that do not contain a 0 surrounded by 1’s, or a 1 surrounded by 0’s. (i.e., all the words that do not contain “010” or “101” as a subword.)

Notice: here you are allowed to use the fact that all languages in Problem 3 are regular, even if you didn’t solve that problem. However, these are the only languages that you are allowed to use, together to the closure properties of regular languages under union, complement, concatenation and star.