1. Construct DFAs that recognize the following languages over the alphabet \{a, b\}:
   1. \{w \mid \text{w contains at least two } a\text{s and at most one } b\}
   2. \{w \mid \text{w does not contain the substring } bba\}
   3. \{w \mid \text{every odd position of } w \text{ is a } b\}
   4. \{w \mid \text{w contains at least three } b\text{s}\}
   5. \(D = \{w \mid \text{w has an equal number of occurrences of } ab \text{ and } ba \text{ as substrings}\}\). Thus \(bab \in D\) because \(bab\) contains a single \(ab\) and a single \(ba\), but \(baba \notin D\) because \(baba\) contains two \(bas\) and one \(ab\).

   Draw the state diagram for all; add a formal description of the DFA for 2), and add a proof that your DFA recognizes exactly the specified language for 4). To do this, you will need to prove that (1) your DFA accepts all strings in the language and (2) any string accepted is in the language.

2. For the NFA in Figure 1, convert it to a DFA using the method of Theorem 20 from the notes.

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

3. Give regular expressions generating the languages of Exercise 1.
4. For each of the following give two strings that are members of the language, and two strings that are not members of the language. Assume the alphabet $\Sigma = \{0, 1\}$.

1. $0^*1^*$.
2. $1(01)^*0$.
3. $\Sigma^*0\Sigma^*1\Sigma^*0\Sigma^*$.
4. $(0110^* \cup 0 \cup 10)^*$.
5. $1^* \cup 0^*$.

5. Obtain the regular expression that describes the same language as the automata in Figure 2, using the method of Lemma 28 from the notes.

![Figure 2: NFA for Problem 5.](image)

6. Prove that every NFA can be converted to an equivalent one that has a single accept state.

7. Prove that the following languages over $\{0, 1, 2\}$ are not regular:

1. $\{0^n1^m2^n \mid n \geq 0\}$
2. $\{0^{p-1} \mid p \text{ is a prime}\}$
3. $\{0^n1^m0^n \mid n, m \geq 0\}$
4. $\{0^n1^m \mid n \neq m\}$