1. Suppose we are given the following NFA over the alphabet \{a, b\} (5 points):

(a) Find a regular expression for the language accepted by the NFA.
(b) Write down the transition table for the NFA.
(c) Use (11.8) (slide 7 of my notes: Regular Language & Finite Automata III) to transform the NFA into a DFA.

2. Transform each of the following NFAs into a DFA (5 points).

(a) \( T(0, a) = \{0, 1\} \), where 0 is start and 1 is final.
(b) \( T(0, a) = \{1, 2\}, T(1, b) = \{1, 2\} \), where 0 is start and 2 is final.

3. Let the set of states for a DFA be \( S = \{0, 1, 2, 3, 4, 5\} \), where the start state is 0 and the final states are 2 and 5. Let the equivalence relation on \( S \) for a minimum-state DFA be generated by the following set of equivalent pairs of states: \( \{(0, 1), (0, 4), (1, 4), (2, 5)\} \). Write down the states of the minimum-state DFA (5 points).

4. For the given regular expression, start by writing down the NFA obtained by algorithm (11.7) (slide 23 of my notes: Regular Language & Finite Automata II). Then use (11.8) to transform NFA into a DFA. Then use (11.10) (slides 21-28 of my notes: Regular Language & Finite Automata III) to find the minimum-state DFA: \((a + b)^*a\) (5 points).

5. Find a regular grammar to describe the following languages (5 points):

(a) \( \{a, aaa, aaaaa, \ldots, a^{2n+1}, \ldots\} \).
(b) \( \{\Lambda, a, b, c, aa, bb, cc, \ldots, a^n, b^n, c^n, \ldots\} \).

6. Any regular language can also be defined by a grammar with productions of the following form, where \( w \) is a nonempty string of terminals: \( S \to \Lambda, S \to w, S \to T \) or \( S \to Tw \). Find a grammar of this form for the language of the given regular expression: \((ab)^*a\) (5 points).
7. Use (11.12) (slide 17 of my notes: Regular Language & Finite Automata IV) to construct an NFA to recognize the language of the given regular grammar: \( S \rightarrow aI \mid bJ, \ I \rightarrow bI \mid \Lambda, \ J \rightarrow aJ \mid \Lambda \) (5 points).

8. Show that the given language is not regular by using the pumping lemma (11.13) (slide 20 of my notes: Regular Language & Finite Automata IV): \( \{a^n b^k \mid n, k \in N \land n \geq k \} \).