1-4. Let us consider an automaton for recognizing words that contain either no or exactly one copy of the letter C. For simplicity, let us only consider the alphabet consisting of three symbols: A, B, and C. This automaton has 3 states:

- the start state (n) corresponding to no C's; this state is also final,
- the state f corresponding to words that have exactly one C; this state is also final, and
- the sink state si corresponding to words that have more than one C's. This state is not final.

The transitions are as follows:

- from the state n, C leads to f, A and B lead back to n;
- from f, A and B lead to f, while C leads to si; and
- from si, every symbol leads to si.

1. Trace, step-by-step, how this finite automaton will check whether the following two words are accepted:

- the word CBA (which this automaton should accept) and
- the word CBAC (which this automaton should reject).

2. Apply the general algorithm to the above tracing to find the parts x, y, and z of the word CBA corresponding to the Pumping Lemma. Check that the "pumped" word xyyz will also be accepted by this automaton.

3. Write down the tuple <Q, Σ, δ, q₀, F> corresponding to this automaton:

- Q is the set of all the states,
- Σ is the alphabet, i.e., the set of all the symbols that this automaton can encounter;
- δ: Q x Σ → Q is the function that describes, for each state q and for each symbol s, the state δ(q, s) to which the automaton that was originally in the state q moves when it sees the symbol s (you do not need to describe all possible transitions this way, just describe two of them);
- q₀ is the staring state, and
• F is the set of all final states.

4. Use a general algorithm that we had in class to generate a context-free grammar corresponding to this automaton. Show how this grammar will generate the word CAB.

5. Let $A_1$ be the automaton described in Problems 1-4. Let $A_2$ be an automaton that accepts all the strings that contain at least one A. This automaton has two states: the start state and the sink state which is also final. The transitions are as follows:

• from the start state, B or C lead back to the start state, A leads to the final state;
• from the final state, any symbol leads back to this state.

Use the algorithm that we had in class to describe the following two new automata:

• the automaton that recognizes the union $A_1 \cup A_2$ of the two corresponding languages, and
• the automaton that recognizes the intersection of the languages $A_1$ and $A_2$.

6-7.

6. Use the general algorithm that we learned in class to design a non-deterministic finite automaton that recognizes the language $(A \cup B)^*C$:

• first, describe the automata for recognizing A, B, and C;
• then, combine them into the automata for recognizing the union $A \cup B$ and the Kleene star $(A \cup B)^*$;
• finally, combine the automata for $(A \cup B)^*$ and C into an automaton for recognizing the desired concatenation of the two languages.

7. Use the general algorithm to transform the resulting non-deterministic finite automaton into a deterministic one.

8-9. Use a general algorithm to transform the finite automaton $A_2$ from Problem 5 into the corresponding regular expression.

10. Prove that the language L of all the words that have equal number of As and Bs and at most one C is not regular.