# Regular Language Library

Write a library for working with regular languages represented by finite automata (DFA and ε-NFA) and/or regular expressions. Implement the following functionalities for working with regular languages:

• Execution of finite automata (DFA and ε-NFA) formed by the library user

• Construction of representations of union, intersection, difference, concatenation of languages, complement of language and the result of applying Kleene star operation to the language, with support for chaining operations

• Determination of the lengths of the shortest and longest words in the language, as well as testing the finiteness of the language

• Minimization of the number of states for deterministic finite automata

• Transformation of ε-NFA into DFA, as well as transformation of a regular expression into a finite automaton

• Comparison of representations of regular languages, including regular expressions, for language equality

Write an application that loads the specification of the representation of a regular language (supported representations are DFA, ε-NFA, and regular expression) and tests the membership of specified strings in the represented language. Perform lexical analysis of the specification of the representation of a regular language and, in case of lexical irregularities, record the number of relevant lines of the specification that contain irregularities.

Write an application for generating source code for simulating a state machine based on the specification of a deterministic finite automaton. Allow users of the generated code to specify reactions to events of entering and leaving a state for each formed state. Allow users of the generated code to specify the reaction to executing a transition for a symbol in the alphabet of the automaton, which can depend on the state from which the transition is made. It should be possible to chain reactions so that an effective chain of reactions to an event can be formed.