Lab 4

Link Github: https://github.com/913-groza-vlad/Formal-Languages-and-Compiler-Design/tree/main/Lab4

FiniteAutomaton class is used for representing a FA, containing the fields:
- states: set of Strings
- alphabet: set of Strings
- transitions: a collection that maps a pair of two strings (the source state and a symbol from the alphabet) and a set of Strings, which represents the destination states corresponding to the key
- initialState: a String
- finalStates: a set of Strings
- a filename, so that the FA will be read from a file by calling the readFAFromFile():
States are read from the first line of the FA.in file and they are separated by space. On the second line, we have the alphabet, each character of the alphabet also being separated by a space. Then, on the third line, there is placed the initial state and from the fourth line read we can retrieve the final states, so that we can add it to our set. Each one of the next lines contains a transition, in the following format: first, there is the source state, then we have a character from alphabet and last the destination state, separated by a space character. Transitions that appear multiple times or contain invalid characters will be ignored.

The FiniteAutomaton class also contains methods for representing each of its fields in string format and the entire FA read from file, but also the methods:

public boolean isDFA():
This method verifies if the read FA is a deterministic finite automaton, by checking in the transitions map if a set associated to a key contains more than one value, i.e. there are no multiple arrows of the same label starting from a state.

public boolean checkSequence(String sequence):
It checks whether the sequence can be successfully processed by the FA. If the input sequence is empty, the method checks whether the initial state is one of the final states. If so, it returns true. This accounts for the case where the FA accepts

the empty string. For each symbol in the input sequence, the method simulates state transitions based on the FA's transition function. It uses a Pair<String, String> (representing the current state and the input symbol) to check if there is a corresponding transition in the transitions map. If a valid transition is found, the current state is updated to the next state specified by the transition. If no transition is found for a symbol, the method immediately returns false since the sequence is not accepted. After processing the entire sequence, the method checks whether the final state reached is one of the FA's final states.

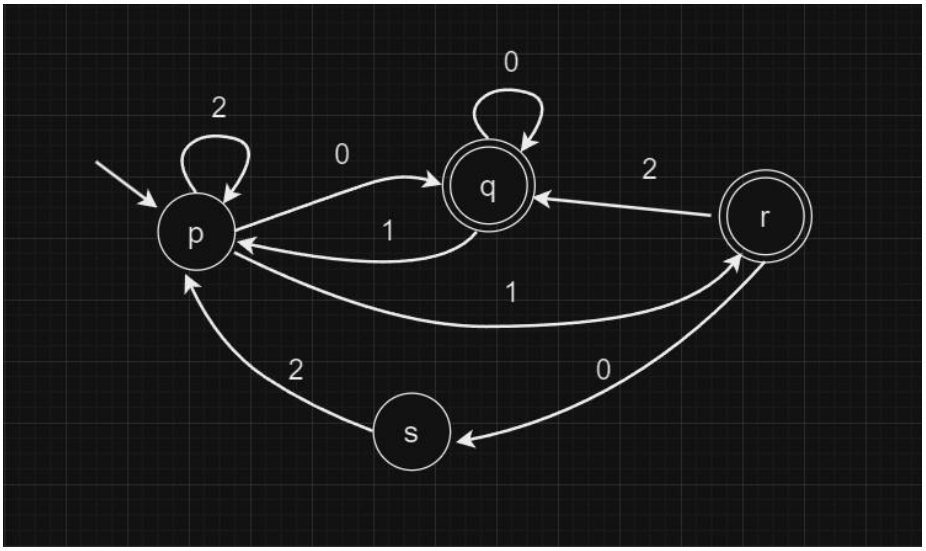
FA should be written in the following format from file:

letter = "a" | "b" | ... | "z" | "A" | "B" | ... | "Z"
digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

alphabetSymbol = letter | digit
state = letter
transition = state " " alphabetSymbol " " state
states = state {" " state}
alphabet = alphabetSymbol {" " alphabetSymbol}
initialState = state
finalStates = state {" " state}
transitions = transition {"\n" transition}

FA = states "\n" alphabet "\n" initialState "\n" finalStates "\n" transitions

Example of a DFA read from file:



- the sequence 10220 is accepted by the FA
- the sequence 2001 is not accepted, because the last state reached is p, which is not part of the set of final state
- the sequence 00221 as there is no transition from state q and the symbol 2