Github link:

https://github.com/x64alex/Compilers/tree/main/Lab4

class DFA:

- A class for the finite automata which has as class fields all states, alphabet, final_states represented as a set, transitions represented as a hasmap(from state and symbol as key and to value as key) and the initial_state as a string.
- The method is_accepted(sequence) checks if the given sequence is accepted by the automata. The function returns true if it is accepted and false otherwise. It does that by starting with the initial state and iterating over the sequence to check for each if there is a state and current symbol there. If it is it goes to the transition value and if not it returns false. At the end, the state we arrived at is verified if it is a part of the possible final states and returns the response.

def read_fa_from_file(file_path):

- A function that reads the content of the input file which has the file_path given and initializes and returns the dfa with the values read from the file. The input file must be in the ebnf defined below. The reading from file is done using split method from Python.

def menu_function():

- A function that displays a menu with changing file option, display fields of dfa and checking if a value is_accepted.

EBNF for input file:

non_zero_digit = 1|2| .. |9
digit = 0|1|..|9
number = non_zero_digit{digit}
letter = a|b|..|z|A|B..|Z
character = letter | digit
range_int = number “-” number
range_string = letter “-” letter
range = range_int | range_string
value = character | range
firstLine = "states" "=" "{" {value} {"," value} "}"
secondLine = {value}
thirdLine = "out_states" "=" "{" {value} {"," value} "}"
fourthLine = "alphabet" "=" "{" {value} {"," value} "}"
triple = "(" {value} "," {value} "," {value} ")"
fifthLine = "transitions" "=" "{" triple {"; " triple} "}"
inputFile = firstLine "\n" secondLine "\n" thirdLine "\n" fourthLine "\n" fifthLine