Github link: https://github.com/PopFelix/LFTC/tree/main/Lab3

I represented the Finite Automata in a class in which I am reading the FA.in file in the constructor, where we have the following elements in EBNF form:

- non-zero-digit = “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9 “

- zero-digit = “0”

- digit = zero-digit | non-zero-digit

- alpha-character = “a” | “A” | …. | “z” | “Z”

- underscore = “_”

- non-digit = underscore | alpha-character

- char = non-digit | digit

- set = “{“ {char} {“,” char } “}”

- states-line = “states” “=” set

- in-state-line = “in_state” “=” {char}

- out-state-line = “out_states” “=” set

- alphabet-line = “alphabet” “=” set

- 3-tuple = “[“ char “:” char “:” char “]”

- transition-line = “transitions” “=” “{“ 3-tuple { “,” 3-tuple } “}”

- fa-file = states-line “\n” in-state-line “\n” out-state-line “\n” alphabet-line “\n” transition-line

Using these rules, I read the file and store all the fields in their corresponding attributes.

I also have some functions that help print all the attributes, and also a checker that verifies if a sequence of characters is accepted by the FA. In the checker, we always start from the initial state and check if the value of the node to the next leaf is corresponding to the value from the sequence until we reach the end of the sequence.