Finite automata parser

-transitions – list containing tuples representing the transitions for the automata

-alphabet – all the characters of the language

-states – list containing the states for the automata

-initial\_states – list containing the initial states for the automata

-final\_states – list containing the final states for the automata

-verify(input, state):

-recursively verifies if the given input can pass through the automata

-params:

-input: string representing the input to be verified

-state: state that the input is in at the current step of the verification

-return: True if the input matches, False otherwise

**Identifiers**

**-a sequence of letters and  digits, such that the first character is a letter; the rule is:**

**<identifier> ::= <letter> | <identifier><letter> | <identifier> <digit>**

**<letter> ::= A | B | ... | Z | … | a | … | z**

**<digit> ::= 0 | 1 |...| 9**

Constants

**1.integer :**

**<numconst> ::= +<num> |-<num> |<num> | 0**

**<num> ::= <non-zero-digit> | <num> <digit>**

**<non-zero-digit> ::= 1 | 2 | … | 9**

**2.character**

**<symbol> = . | space | - | ? | ! | , | ‘ | “ | ;**

**<character> ::= <letter>|<digit>|<symbol>**

**3.string**

**<string> ::= <character> | <string> <character> | empty**

[GitHub - TudorFernea/FLCDLab2](https://github.com/TudorFernea/FLCDLab2)