Finite Automata – Documentation
Turcas Andrei 937/2

GitHub repository: https://github.com/andrewj77/FLCD

Implementation of Finite Automata -> it is implemented as a Python class, having 5 relevant fields: initial_state – character, final_states, all_states, alphabet – lists, transitions – dictionary of elements that have as key a pair of a state and a symbol, and as value the transition state.

The input file that is read and interpreted by the Finite Automata Class should have the following structure:
Q = {" ", "state"}
∑ = {" ", "input_symbol"}
q0 = "state"
F = {" ", "state"}
δ = {"\n", "(", "state", ",", "input_symbol", ")", " ", "->", " ", "state"}
Example:

```
Q = A B C
Σ = 0 1
q0 = A
F = A C
δ =
(A,0) -> A
(A,1) -> B
(B,0) -> A
(B,1) -> C
(C,1) -> C
```

Checking if a sequence is accepted by the Finite Automata:

- Before starting the actual check, we verify that the FA is deterministic, if it is not we do not pursue the operation
- Initially we set the current state as the initial state
- For each character of the sequence, we check if the pair of the current state and the character of the sequence is an existent key in the transitions dictionary
- After successfully checking each character, we verify that the current state is among the final ones

The regular expressions for detecting identifiers and constants were replaced by 2 Finite Automatons. Now we check if the corresponding token is an accepted sequence in one of the two defined automatons. They can be found on GitHub, in 'identifier_fa.txt' and 'constant_fa.txt'.