

Link to GitHub repository: <https://github.com/octaviaah/flcd>

The Finite Automata is structured as a class with 5 fields:

- Q, which represents the set of states
- E, which represents the alphabet of the language
- q0, which represents the initial state
- F, which represents the set of final states
- S, which represents the set of state transitions. A HashMap is used to represent S in code

The process of checking is a Finite Automata is DFA:

DFA refers to Deterministic Finite Automation. A Finite Automata(FA) is deterministic iff to an input symbol, which belongs to the alphabet, corresponds a single state. In shorter words, an FA is a DFA if there is only one transition. Otherwise, an FA is Non-deterministic Finite Automation(NFA).

To check if an FA is DFA, we are looping through the keys of S and check if there is a list with more than one element.

The process of checking if a sequence is accepted by the FA:

To check if the sequence is accepted by the FA, we loop through the symbols of the sequence, and check that the sequence can be reach following the transitions.

Integration with the Scanner lab:

The matching regex of identifiers and constants is replaced with an FA that respects the constraint to which a sequence is an identifier or a constant. The FAs are kept in 2 files(identifiers.in and constants.in) and are read before the process of tokenizing starts.