<https://github.com/alexandra-olahut/FLCD-assignments>

FLCD Documentation

Olahut Alexandra – 935/2

Finite Automata

**Finite automata**

Was implemented as a class with the following fields:

* *states*: list of states
* *alphabet*: list of alphabet symbols
* *initial*: the initial state
* *finals*: list of final states
* *transitions*: represented with a dictionary where the keys are (source\_state, symbol) and the value is the target state / list of states

The FA is read from a file with the following structure:

**file** = states *NEWLINE* alphabet *NEWLINE* initial *NEWLINE* finals *NEWLINE* transitions

**states** = state {*SPACE* state}

**alphabet** = symbol {*SPACE* symbol}

**initial** = state

**finals** = state {*SPACE* state}

**transitions** = transition {*NEWLINE* transition}

**transition** = state *SPACE* symbol *SPACE* state

**state** = letter {digit}

**symbol** = letter | digit | specialCharacter

**letter** = “a” | … | “z” | “A” | … | “Z”

**digit** = “0” | “1” | … | “9”

**specialCharacter** = “+” | “\_” | “-“ | …

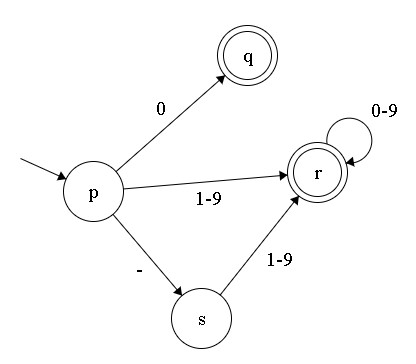
**FA methods**:

* *read\_from\_file*: initializes the fields read from a given file
* validate: checks if the FA is valid, verifying some conditions (initial and final states are among the states, a transition is formed as (state,symbol)->state, states and symbols don’t intersect)
* *isDFA*: checks if the finite automata is deterministic
* *isSequenceAccepted*: for a given sequence (if the FA is DFA) checks if the sequence is accepted by the FA
* \*getters for all the fields

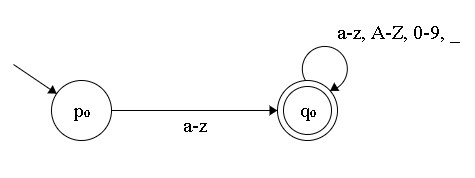
**FA integration in Scanner**

Two finite automatas are used in the scanner, one for integer constants and one for identifiers, each read from its separate file (*FA\_id.in, FA\_int.in*).

***Integers***

******

***Identifiers***

******