

Documentation

Moldovan Andrei

Github link: <https://github.com/andreimoldovan23/FLCD>

The FA contains the following fields:

- states (representing all possible states)
- events (representing the alphabet)
- initial state (the initial state)
- final states (all final states)
- transitions (they are of the form $(x, y) \rightarrow z$, where x and z are states and y belongs to the alphabet); the internal representation of this takes the form of a list of `Pair<Pair<String, String>, String>`

When the FA class is constructed, it will read and parse the fa file and store data accordingly.

This is when the check for DFA takes place. If any transition has more than one state to which it leads the program will throw an error.

For checking if a given sequence is accepted, the program will first split this sequence into characters and then, for each character, it will check if it is part of the alphabet. If not, it will throw an error. Then it will try to transition from the current state with the current character according to the internal list of transitions. If there is no such transition the program will throw an error.

At the end it will check if the state that we've reached is indeed a final one. If not, it will throw an error.

FA.in example

- on the first line, separated by space are all the states
- on the second line, separated by space, are all symbols in the alphabet
- on the third line is the initial state
- on the fourth line, separated by space, are all the final states

- on the following lines, separated by space, are the elements which make up each transition

```
p q r
a b
p
r
p a p
p b q
q b q
q a r
r a r
```

For this example the values are the following:

- states: p, q, r
- alphabet: a, b
- initial state: p
- final states: r
- transitions: {
 - (p, a) -> p
 - (p, b) -> q
 - (q, b) -> q
 - (q, a) -> r
 - (r, a) -> r}