Write a program that:

1. Reads the elements of a FA (from file)

2. Displays the elements of a finite automata, using a menu: the set of states, the alphabet, all the transitions, the set of final states.

3. For a DFA, verify if a sequence is accepted by the FA.

Deliverables:

1. FA.in - input file (*on Github*)
2. Source code (*on Github*)
3. Documentation. It should also include in BNF or EBNF format the form in which the FA.in file should be written (*on Moodle and Github*)

The git link: https://github.com/DeeaKr/FLCD/tree/develop/lab4

the FA from FA.in : $M = (Q, \Sigma, \delta, q_0, F)$, $Q = \{ q_1, q_2, q_3, q_4\}$, $\Sigma = \{a, b, c\}$, $F = \{q_3, q_4\}$,

δ	a	b	c
q_1	$\{q_1, q_2\}$	$\oslash$	$\oslash$
q_2	$\oslash$	$\{q_2, q_3\}$	$\oslash$
q_3	$\oslash$	$\oslash$	$\{q_4\}$
q_4	$\oslash$	$\oslash$	$\oslash$

Representation

The FA class contains:

1. A set of strings for all the states

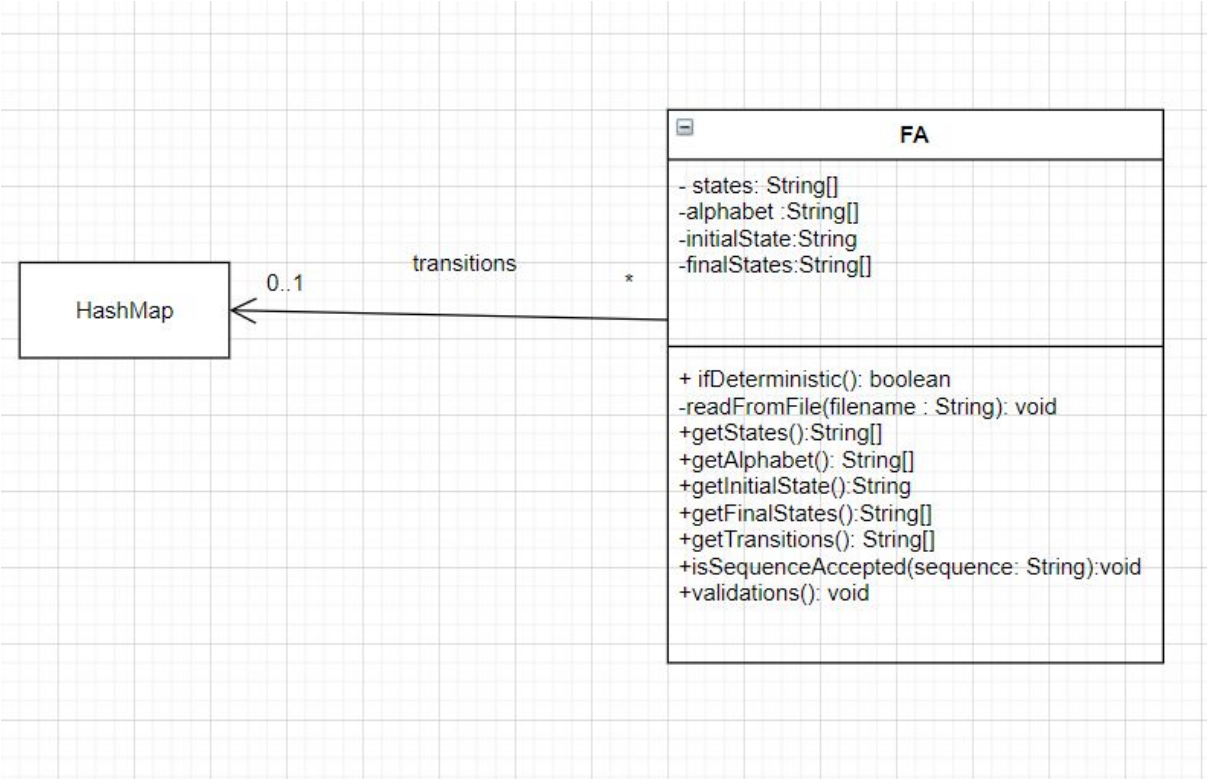
2. A set of strings for the alphabet

3. A String for the initial State

4. A set of string for the final states

5. A HashMap with a pair<String,String> as a key and a list of Strings as a value, for the transitions.

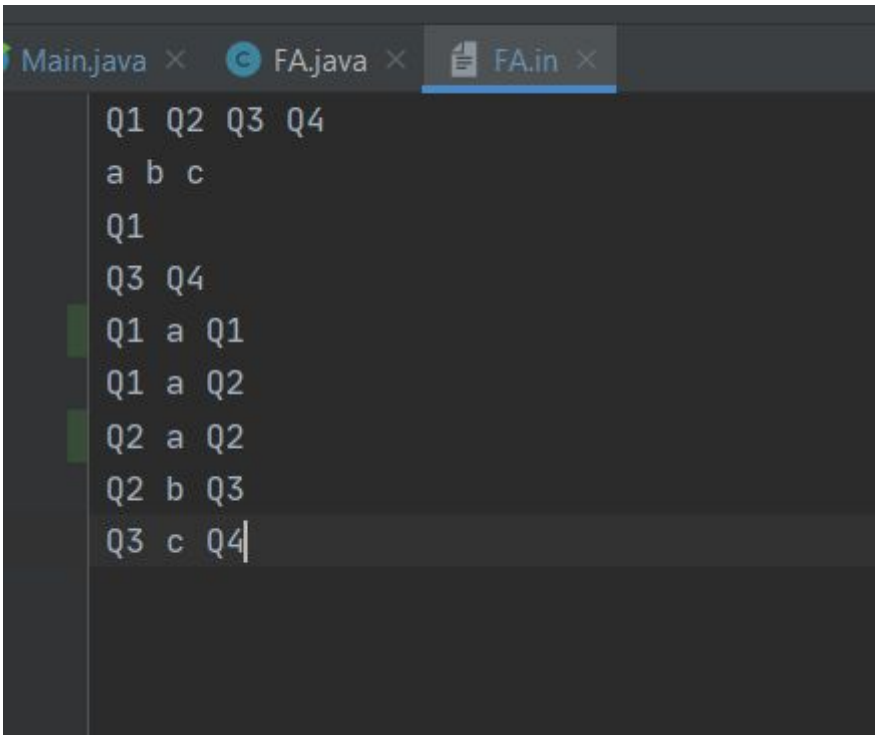
Class diagram:



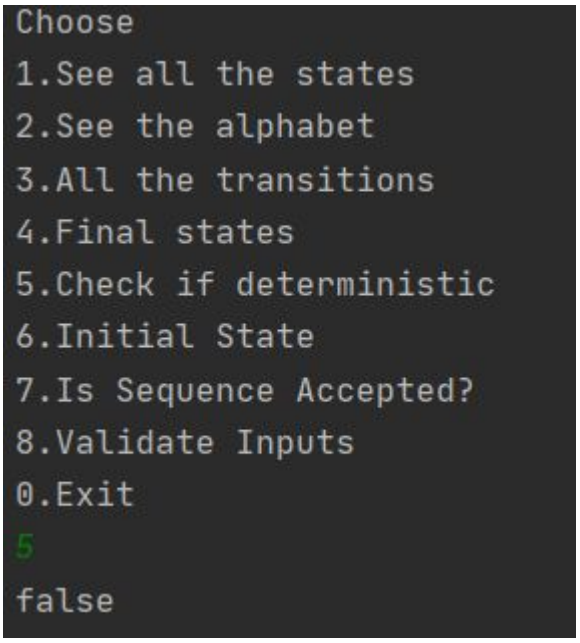
The function ifDeterministic checks if the list of Strings from the hashmap transitions is equal or smaller than 1. Returns false if the list is greater than 1, true otherwise.

```java
public boolean isSequenceAccepted(String sequence){
    if(ifDeterministic()) {
        String current = initialState;
        int i = 0;
        boolean found = true;
        while (i < sequence.length() && found) {
            found = false;
            for (Pair<String, String> keys : transitions.keySet()) {
                if (keys.getKey().equals(current) && keys.getValue().equals(String.valueOf(sequence.charAt(i)))) {
                    current = transitions.get(keys).get(0);
                    found = true;
                }
            }
            i++;
        }
        return finalStates.contains(current) && i == sequence.length();
    }
    else{
        System.out.println("Is not deterministic");
        return false;
    }
```

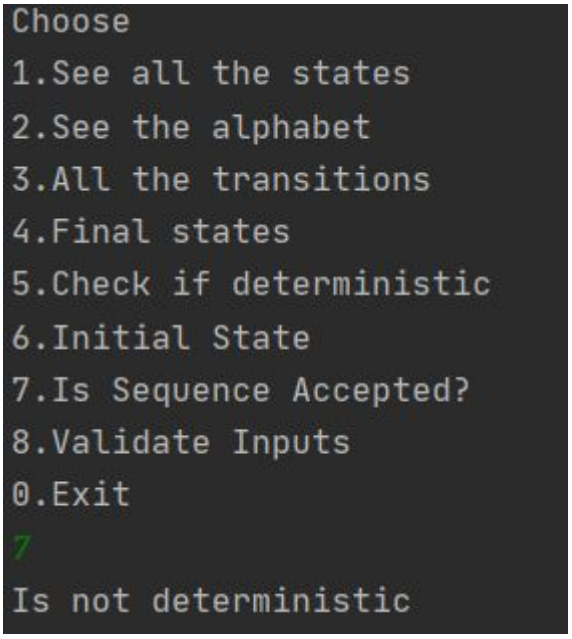
The function isSequenceAccepted receives a String as parameter and returns false if the sequence is not accepted and true otherwise. First we keep a current state which starts from the initial state, we look for the current state along with the string at position i, if we find this pair we go to the next state and change the boolean to true. At the end we check if the current state is between the finals states and if we went through all the string.



When we test this and chose number 5 from menu



We receive this output. If we choose 7 we will receive this message:



When we have in FA.in this input:

Main.java | FA.java | FA.in

```
Q1 Q2 Q3 Q4
a b c
Q1
Q3 Q4
Q1 a Q2
Q2 b Q3
Q3 c Q4

```

And we choose nr 5 from menu we receive:

```
Choose
1.See all the states
2.See the alphabet
3.All the transitions
4.Final states
5.Check if deterministic
6.Initial State
7.Is Sequence Accepted?
8.Validate Inputs
0.Exit
5
true
```

If we choose nr 7 with from menu with this sequence

```
case 7:
    System.out.println(fa.isSequenceAccepted("abc"));
    break;
```

We receive this :

```
Choose
1.See all the states
2.See the alphabet
3.All the transitions
4.Final states
5.Check if deterministic
6.Initial State
7.Is Sequence Accepted?
8.Validate Inputs
0.Exit
7
true
```

And if we have this sequence :

```
    case 7:
        System.out.println(fa.isSequenceAccepted("b"));
        break;
```

We receive:

```
Choose
1.See all the states
2.See the alphabet
3.All the transitions
4.Final states
5.Check if deterministic
6.Initial State
7.Is Sequence Accepted?
8.Validate Inputs
0.Exit
7
false
```

.