Practica 2

Tomas Barrionuevo Cordoba

October 2022

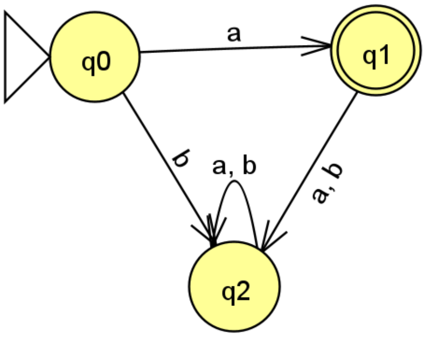
Activities

1. Consider the language over the alphabet a, b that only contains the string a.

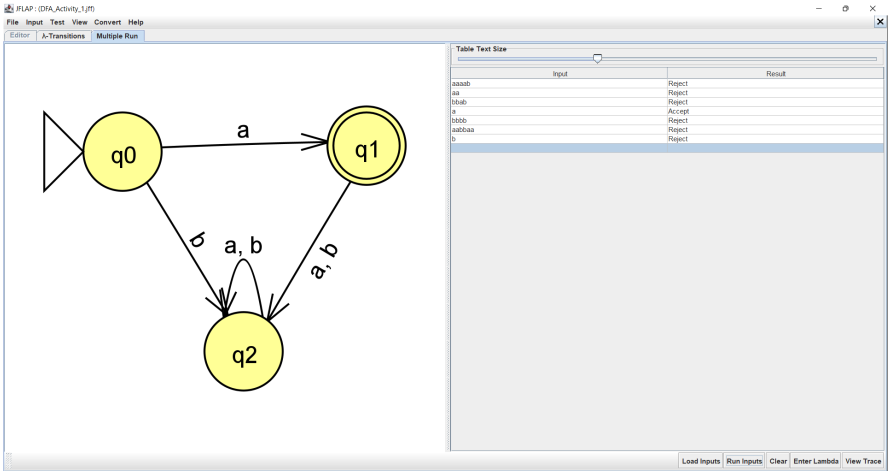
a. Build a DFA that recognizes this language and rejects all those strings that do not belong to the language.

$M = (\{q0, q1\}, \{a, b\}, q0, \{(q0, a, q1), (q0, b, q2), (q1, a/b, q2), (q2, a/b, q2)\}, q1)$

$K = (q0, q1, q2), \sum = (a, b), S = (q0), \delta = \{(q0, a, q1), (q0, b, q2), (q1, a/b, q2), (q2, a/b, q2)\}, F = (q1)$



b. Test the automaton that you have created by introducing 6 chains.

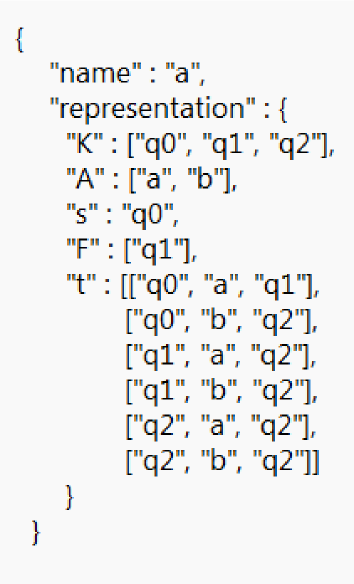


Input	Result
aaaab	Reject
aa	Reject
bbab	Reject
a	Accept
bbbb	Reject
aabbaa	Reject
b	Reject

2. Finite automaton in Octave:

a. Open the Octave finiteautomata.m script and test it with the given example (see script help) in the GitHub repository.

b. Specify in finiteautomata.json the automaton created in Activity 1 and test it with the script!



```
{
    "name" : "a",
    "representation" : {
      "K" : ["q0", "q1", "q2"],
      "A" : ["a", "b"],
      "s" : "q0",
      "F" : ["q1"],
      "t" : [["q0", "a", "q1"],
             ["q0", "b", "q2"],
             ["q1", "a", "q2"],
             ["q1", "b", "q2"],
             ["q2", "a", "q2"],
             ["q2", "b", "q2"]]
      }
  }
```

3. Test the Free Context Pumping Condition for the first three examples

4. Finite automaton in Octave: