

Teoría de Autómatas y Lenguajes Formales

Práctica 2: Autómatas en JFLAP

Lázaro, Vargas García

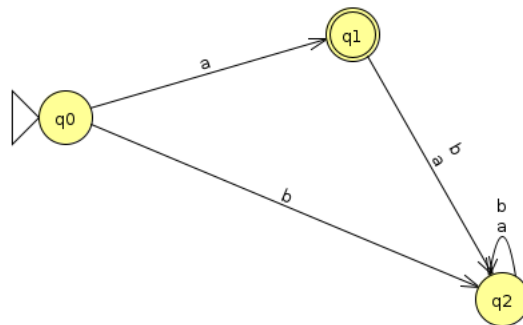
26 de octubre de 2022

1. **Considera el lenguaje $L = \{a\}$ sobre el alfabeto $\{a, b\}$**
- 1.1. **Construye un DFA que reconozca este lenguaje y rechace todas las cadenas que no pertenecen a este**

Vamos a construir el DFA como sigue: $M = (K, \{a, b\}, \delta, s, F)$, Siendo:

- $K = \{q0, q1, q2\}$
- $s = q0$
- $F = \{q1\}$
- $\delta : K \times \Sigma \rightarrow K$ es la función de transición, cuyas imágenes están definidas como sigue:
 - $\delta(q0, a) = q1$
 - $\delta(q0, b) = q2$
 - $\delta(q1, a) = q2$
 - $\delta(q1, b) = q2$
 - $\delta(q2, a) = q2$
 - $\delta(q2, b) = q2$

La representación del DFA como grafo dirigido quedaría de la siguiente forma:



Parece que el autómata está bien especificado, pues de las cadenas introducidas, sólo acepta a .

1.2. Prueba el autómata que has creado introduciendo 6 cadenas

El resultado de la prueba se ve en esta imagen:

The screenshot shows the JFLAP application window titled "JFLAP : (sol.jff)". The "Multiple Run" tab is active. On the left, the finite automaton diagram is displayed. On the right, a table titled "Table Text Size" shows the results of testing six input strings. The table has two columns: "Input" and "Result".

Input	Result
a	Accept
aabb	Reject
babbbb	Reject
bbaa	Reject
aaaa	Reject
b	Reject

At the bottom of the table, there are buttons for "Load Inputs", "Run Inputs", "Clear", and "Enter".

2. Autómata finito en Octave

2.1. Abre el script *finiteautomata.m* en Octave y pruébalo con el ejemplo proporcionado en la ayuda

Para ello, he ejecutado el comando *help finiteautomata.m* y se me proporciona el siguiente ejemplo:

```
octave:i> help finiteautomata.m
'finiteautomata.m' is the file /media/lalovrgas/Archivos/UMA/Tercero/Primer cuatrimestre/TALF/talfuma/software/automata/finiteautomata.m

finiteautomata(automatonname, string, formatoption)

Computation for a given finite automaton and string.
The automaton can be either DFA or NFA, and it is defined
in a JSON file, like this:

{
  "K" : ["q0", "q1", "q2"],
  "A" : ["a", "b"],
  "S" : "q0",
  "F" : ["q2"],
  "t" : [{"q0", "a", "q1"},
          ["q1", "a", "q1"],
          ["q1", "b", "q2"],
          ["q2", "b", "q2"]]
}

(a transition consuming the empty string: ["q1", "", "q2"])

For example:

>> finiteautomata("aa*bb*", "ab")

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

x = ab

(q0, ab) ⊢ (q1, b) ⊢ (q2, ε)

x ∈ L(M)
```

Después, he ejecutado yo mismo el script usando el ejemplo anterior:

```
lalovrgas@lalovrgas-BOHK-WAX9X: /media/lalovrgas/Archiv...
fjv, 02/01/2022 loads from automata database

=====

Additional help for built-in functions and operators is
available in the online version of the manual. Use the command
'doc <topic>' to search the manual index.

Help and information about Octave is also available on the WWW
at https://www.octave.org and via the help@octave.org
mailing list.
octave:2> finiteautomata("aa*bb*", "ab")
warning: strmatch is obsolete; use strncmp or strcmp instead

M = ( {q0, q1, q2}, {a, b}, q0, {q2}, {(q0, a, q1), (q1, a, q1), (q1, b, q2), (q
2, b, q2)} )

w = ab

(q0, ab) ⊢ (q1, b) ⊢ (q2, ε)

x ∈ L(M)
octave:3> 
```

Observemos que la cadena ab es aceptada porque pertenece al lenguaje generado por la expresión regular (aa^*bb^*) .

2.2. Especifica en *finiteautomata.json* el autómata creado en la Actividad 1 y pruébalo con el script

He especificado mi autómata de la siguiente forma:

```
58 {
59   "name" : "miAutomata",
60   "representation" : {
61     "K" : ["q0", "q1", "q2"],
62     "A" : ["a", "b"],
63     "S" : "q0",
64     "F" : ["q1"],
65     "t" : [
66       ["q0", "a", "q1"],
67       ["q0", "b", "q2"],
68       ["q1", "a", "q2"],
69       ["q1", "b", "q2"],
70       ["q2", "a", "q2"],
71       ["q2", "b", "q2"]
72     ]
73   }
74 }
```

El resultado de las pruebas ha sido:

```

octave:4> finiteautomata("miAutomata", "a")

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

w = a

(q0, a) ⊢ (q1, ε)

x ∈ L(M)
octave:5> finiteautomata("miAutomata", "ab")

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

w = ab

(q0, ab) ⊢ (q1, b) ⊢ (q2, ε)

x ∉ L(M)
octave:6>

```

```

octave:6> finiteautomata("miAutomata", "bbaa")

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

w = bbaa

(q0, bbaa) ⊢ (q2, baa) ⊢ (q2, aa) ⊢ (q2, a) ⊢ (q2, ε)

x ∉ L(M)
octave:7> finiteautomata("miAutomata", "bbbb")

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

w = bbbb

(q0, bbbb) ⊢ (q2, bbb) ⊢ (q2, bb) ⊢ (q2, b) ⊢ (q2, ε)

x ∉ L(M)
octave:8>

```

Sólo se acepta la cadena a , como era de esperar.