

FUNDAMENTOS TEÓRICOS DA COMPUTAÇÃO

-- AUTÔMATO FINITO DETERMINÍSTICO --

Definição e exemplos

Descrição informal (1)

Um autômato processa uma **entrada**, que é obtida a partir de um conjunto de **símbolos** vindos de um **alfabeto**

O conjunto de símbolos recebidos pelo autômato como entrada são chamados de **palavras**

Durante a execução o autômato sempre está em um de seus estados

Ao receber uma nova entrada, o autômato se move para outro estado, baseado em uma **função de transição**, que tem o estado atual e o símbolo lido como parâmetros

Uma vez que a palavra de entrada foi totalmente lida, **o autômato deve parar**, aceitando ou não a entrada

Descrição informal (2)

O estado no qual o autômato **para** é chamado de **estado final**

Dependendo do estado final, o autômato pode **aceitar** ou **rejeitar** uma palavra de entrada

Existe um subconjunto de estados do autômato, que é definido como o **conjunto de estados de aceitação**

Se o estado final é **um** estado de aceitação, então o autômato **aceita** a palavra, caso contrário, a palavra é **rejeitada**

O conjunto de todas as **palavras aceitas** por um autômato é chamado de **linguagem reconhecida** pelo autômato

Resumindo, um autômato é um objeto matemático que toma uma **palavra** como entrada e decide se a **aceita** ou **rejeita**

Descrição formal

Um autômato determinístico finito é representado formalmente por uma 5-tupla $(Q, \Sigma, \delta, q_0, F)$

Q é um conjunto finito de estados

Σ é um conjunto finito de símbolos, chamado de alfabeto do autômato

δ é a função de transição, isto é, $\delta: Q \times \Sigma \rightarrow Q$

q_0 é o estado inicial, isto é, o estado do autômato antes de qualquer entrada ser processada, onde $q_0 \in Q$

F é um conjunto de estados de Q (isto é, $F \subseteq Q$) chamado de estados de aceitação

Exemplo

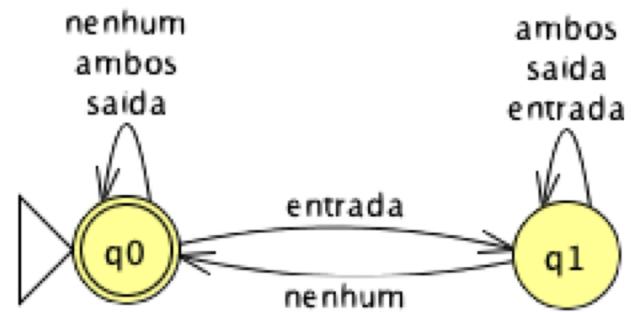
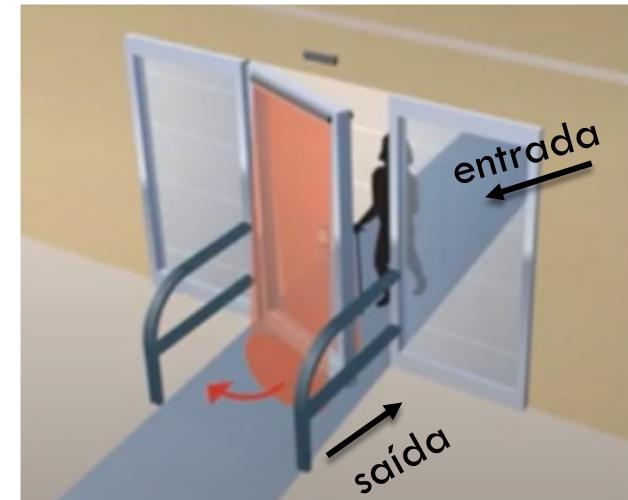
Projeto de um **controlador de porta automática giratória parcial**

Há um **tapete com sensor de presença**, tanto na entrada quanto na saída, para identificar se há pessoas **próximas** da porta

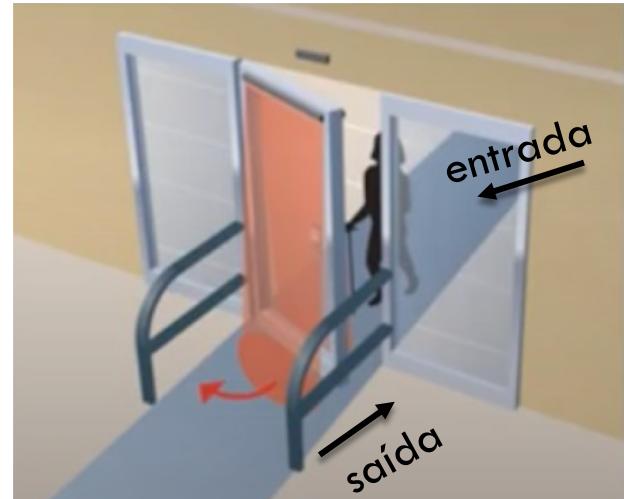
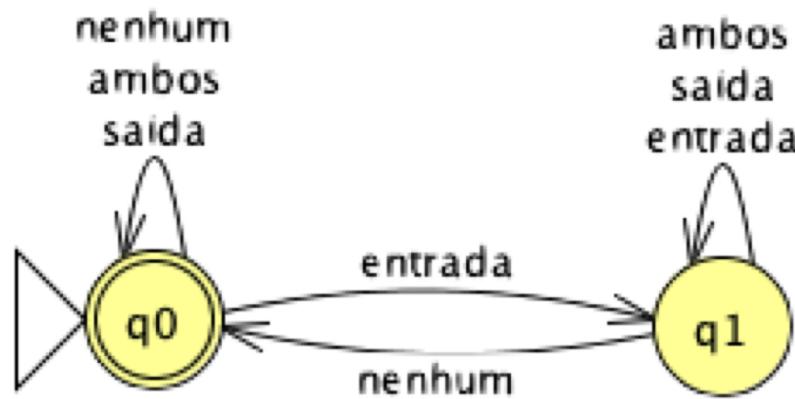
Por motivo de segurança, se tiver alguém no **tapete de saída**, a porta deve ficar **fechada**

Dois estados: porta fechada (q_0) e porta aberta (q_1)

Quatro possíveis entradas: alguém na entrada, alguém na saída, ambos e nenhum



Exemplo



	Nenhum	Entrada	Saída	Ambos
Fechada	Fechada	Aberta	Fechada	Fechada
Aberta	Fechada	Aberta	Aberta	Aberta

Exemplo - JFLAP

File Input Test View Convert Help X

Editor Multiple Run

Table Text Size

Input	Result
saidaentradaentradaambossaidanenhum	Accept
nenhumambosambosaida	Accept
entradaentradaentra	Reject
entradanenhum	Accept
entradanenhumentradanenhum	Accept
ambosnenhumentradaaentradaentradaentr...	Accept

Load Inputs Run Inputs Clear Enter Lambda View Trace

Exemplos

Exemplo 1

$$Q = \{q_0, q_1, q_2\}$$

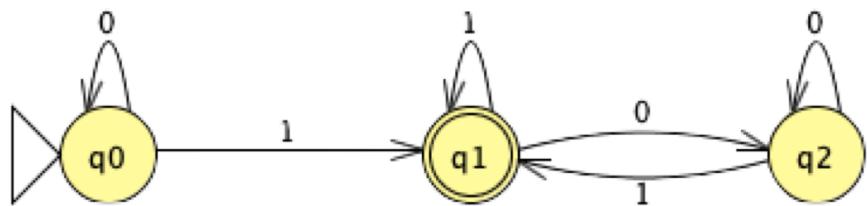
$$\Sigma = \{0, 1\}$$

estado inicial: q_0

$$F = \{q_1\}$$

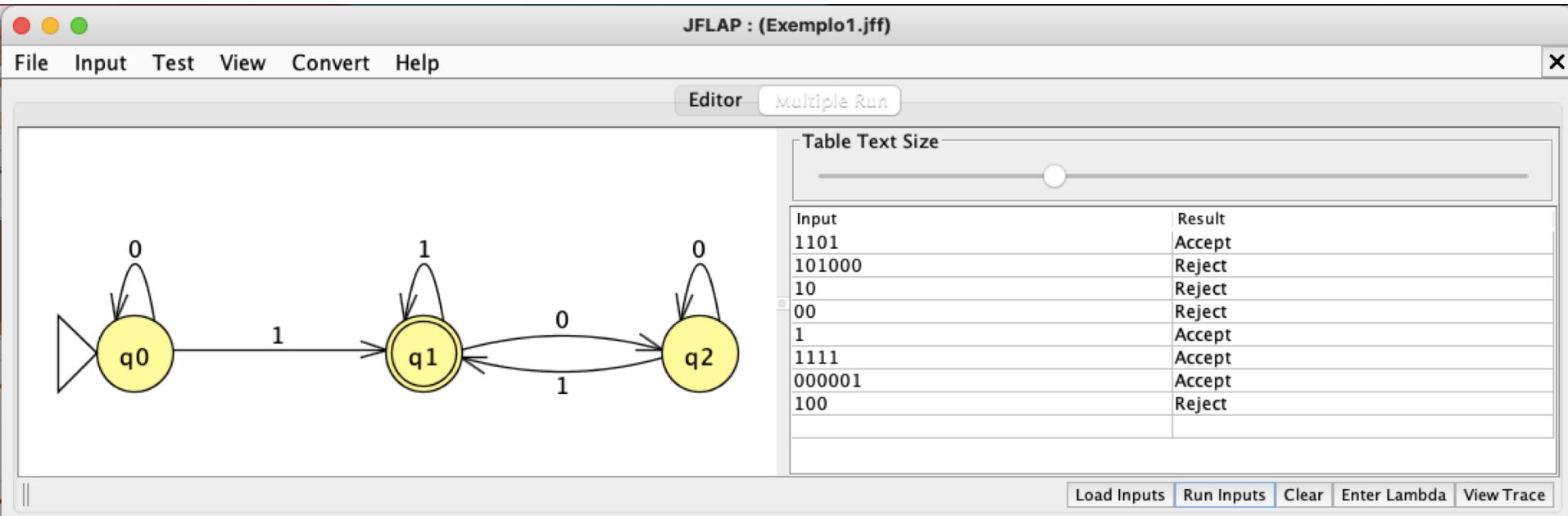
δ é descrita pela tabela

	0	1
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_1	q_1



$$L = \{w \mid w \text{ sempre termina com "1"}\}$$

Exemplo 1

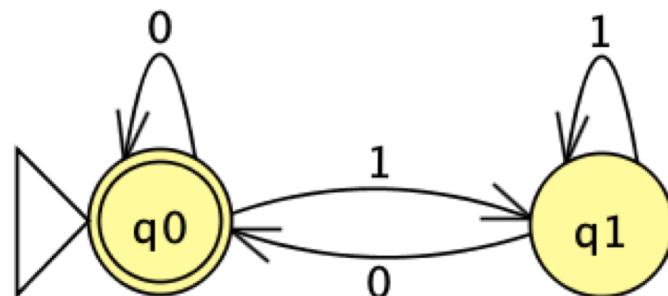


Exemplo 2

$$M = (\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_0\})$$

δ é descrita pela tabela

	0	1
q_0	q_0	q_1
q_1	q_0	q_1



$$L = \{w \mid w \text{ é a cadeia vazia ou termina com "0"}\}$$

Exemplo2

JFLAP : (Exemplo2.jff)

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
1101	Reject
101000	Accept
10	Accept
00	Accept
1	Reject
1111	Reject
000001	Reject
	Accept

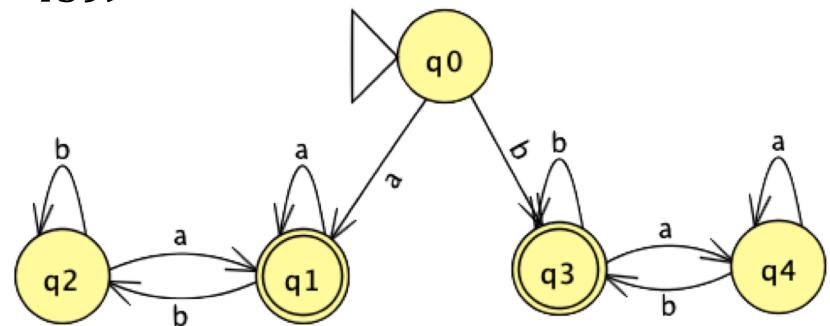
Load Inputs Run Inputs Clear Enter Lambda View Trace

Exemplo 3

$$M = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \{q_1, q_3\})$$

δ é descrita pela tabela

	a	b
q_0	q_1	q_3
q_1	q_1	q_2
q_2	q_1	q_1
q_3	q_4	q_3
q_4	q_4	q_3



$L = \{w \mid w \text{ é a cadeia que, se} \\ \text{começar com "a" tem que terminar} \\ \text{com "a"; se começar com "b" tem que} \\ \text{terminar com "b"}\}$

Exemplo3

JFLAP : (Exemplo3.jff)

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
aaaaa	Accept
bbbbbbb	Accept
ab	Reject
ba	Reject
baaaa	Reject
abbb	Reject
abbbbba	Accept
aaaaaaaaab	Accept
baab	Accept
abbab	Reject

Load Inputs Run Inputs Clear Enter Lambda View Trace

```
graph LR; start(( )) --> q0((q0)); q0 -- a --> q1((q1)); q0 -- b --> q2((q2)); q1 -- a --> q0; q1 -- b --> q2; q2 -- a --> q1; q2 -- b --> q2; q3((q3)) -- sigma --> q0; q3 -- a --> q1; q3 -- a --> q4((q4)); q3 -- b --> q2; q4 -- a --> q3; q4 -- b --> q4;
```

Exemplo 4

Considere alfabeto $\{0,1\}$. Construa um AFD que reconheça a linguagem onde todas as cadeias possuam um número IMPAR de 1s

JFLAP : (Exemplo4.jff)

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

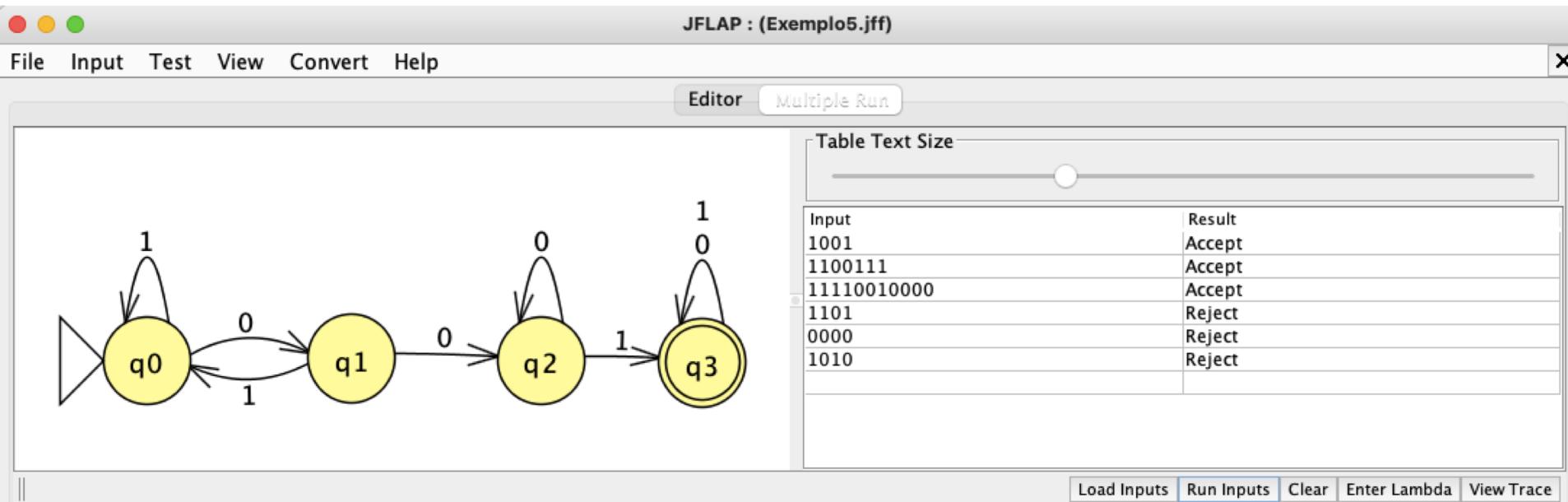
Input	Result
111	Accept
1	Accept
11	Reject
110001	Accept
10101	Accept

```
graph LR; par((par)) -- 0 --> par; par -- 1 --> impar((impar)); impar -- 0 --> impar;
```

Load Inputs Run Inputs Clear Enter Lambda View Trace

Exemplo 5

Considerando alfabeto $\{0,1\}$. Construa um AFD que reconheça a linguagem onde todas as cadeias possuam "001" como subcadeia.



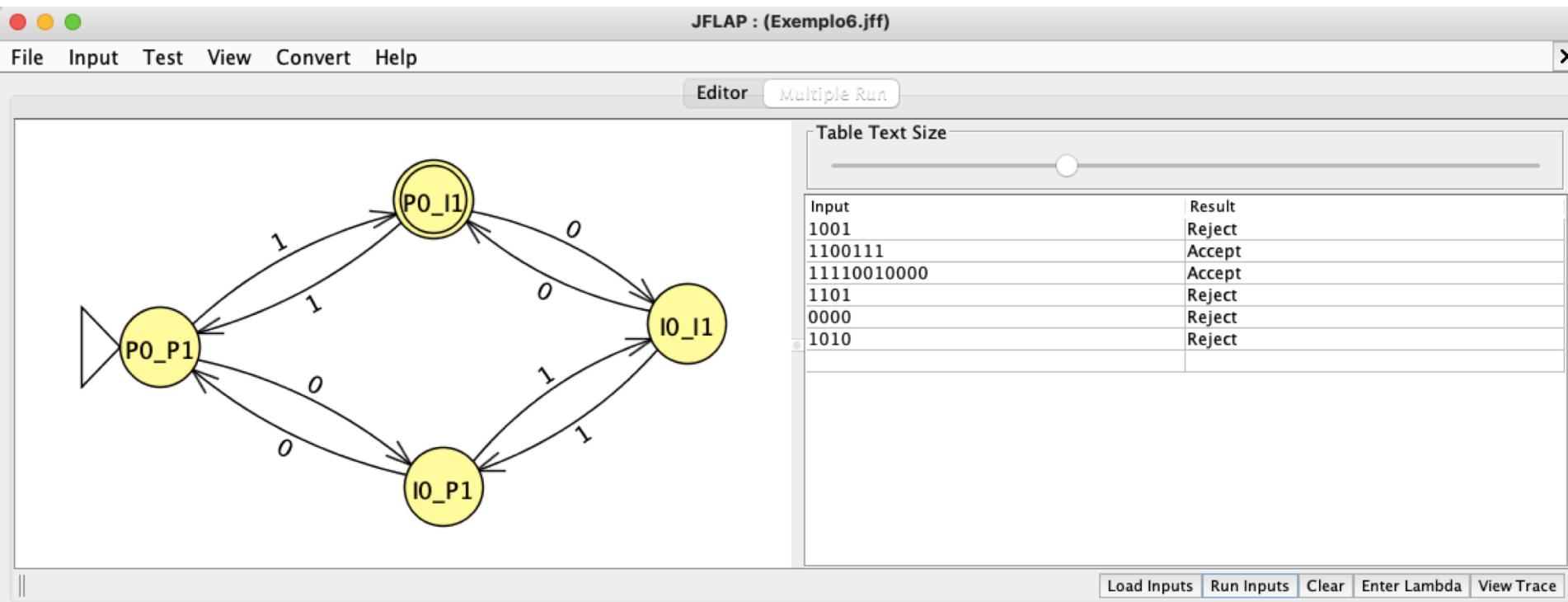
Exemplo 6

Considerando alfabeto $\{0,1\}$. Construa um AFD que reconheça a linguagem onde todas as cadeias possuam um número par de 0's e número ímpar de 1's

Em qualquer cadeia gerada pelos caracteres 0 e 1, temos sempre 4 possibilidades:

- 1) numero ímpar de 0's e ímpar de 1's
- 2) numero ímpar de 0's e par de 1's
- 3) numero par de 0's e par de 1's
- 4) numero par de 0's e ímpar de 1's

Exemplo 6



Exemplo 7

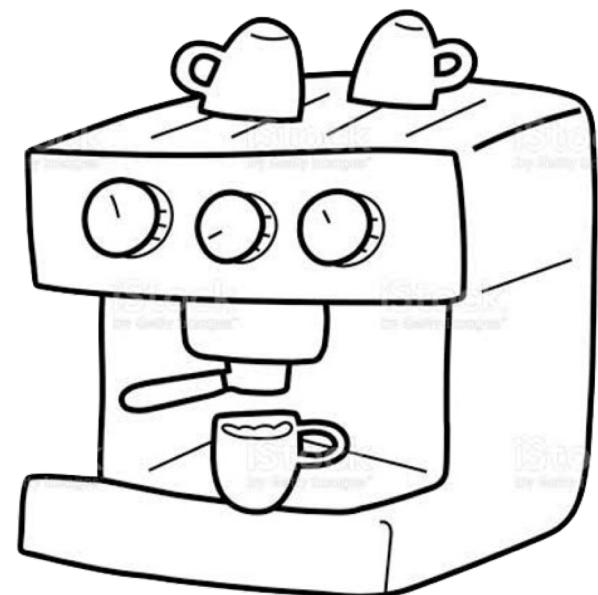
Máquina de Vender Café

A máquina aceita moedas de 5, 10 e 25 centavos

Quando 30 centavos forem inseridos, a máquina chega ao estado de aceitação e o café deve ser entregue

Não é admitido inserir mais do que 30 centavos

A máquina tem conhecimento de quanto dinheiro foi inserido simplesmente através dos estados internos da máquina



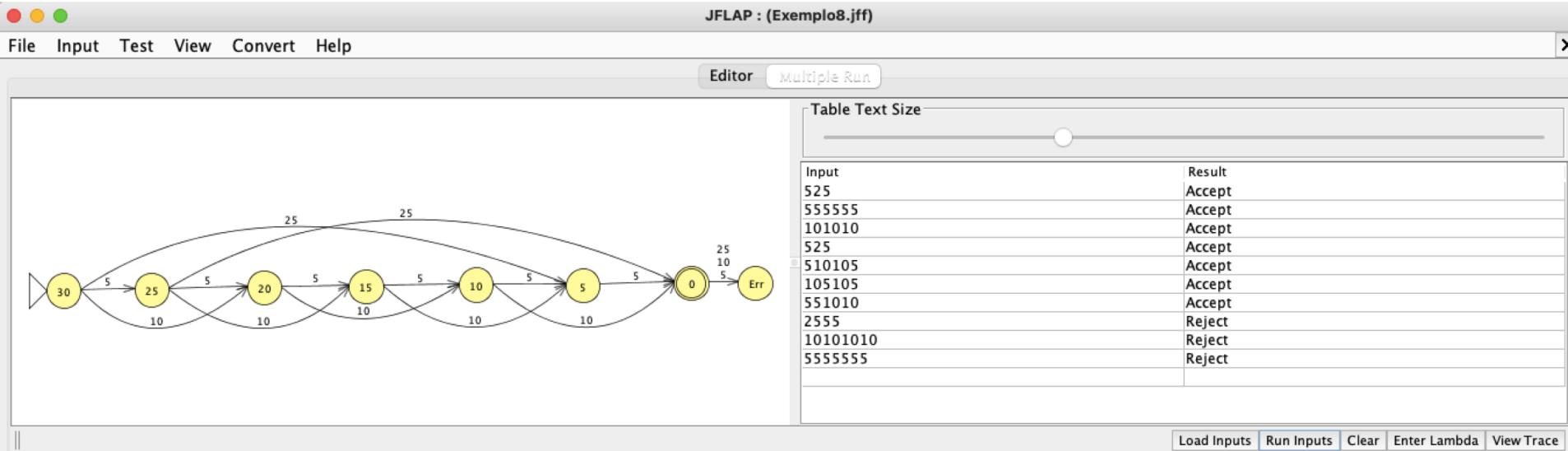
Exemplo 7

Máquina de Vender Café

Cada estado representa uma das seguintes informações:

- faltam 30 centavos para o café ser entregue
- faltam 25 centavos para o café ser entregue
- faltam 20 centavos para o café ser entregue
- faltam 15 centavos para o café ser entregue
- faltam 10 centavos para o café ser entregue
- faltam 5 centavos para o café ser entregue
- café entregue

Exemplo 7



Exemplo 8

$L = \{p \in \{0,1\}^* \mid \text{todo } 0 \text{ em } p \text{ é seguido de } 11\}$

JFLAP : (Exemplo8.jff)

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
11101110111	Accept
011	Accept
10111	Accept
111110011	Reject
101010	Reject

```
graph LR; q0((q0)) -- 0 --> q1((q1)); q1 -- 1 --> q2((q2)); q2 -- 0 --> q3((q3)); q3 -- 0 --> q1; q3 -- 1 --> q3; q0 -- 1 --> q0; q3 -- 1 --> q3;
```

Load Inputs Run Inputs Clear Enter Lambda View Trace

Referências

Newton José Vieira. Introdução aos Fundamentos da Computação. Editora Thompson, 2006.

SIPSER, M. Introdução à Teoria da Computação. São Paulo: Thomson Pioneira, 2007

MENEZES, P. B. Linguagens Formais e Autômatos. 6 ed. Porto Alegre: Artmed, 2011.

<https://www.jflap.org/tutorial/>

<https://www.jflap.org/jflapbook/jflapbook2006.pdf>



PROGRAMMERS

**I THOUGHT YOU
WERE SMART!**

COMPILER

**I'M SOFTWARE, NOT
A MIND READER!**