ACH2043
INTRODUÇÃO À TEORIA DA COMPUTAÇÃO

Aula 4

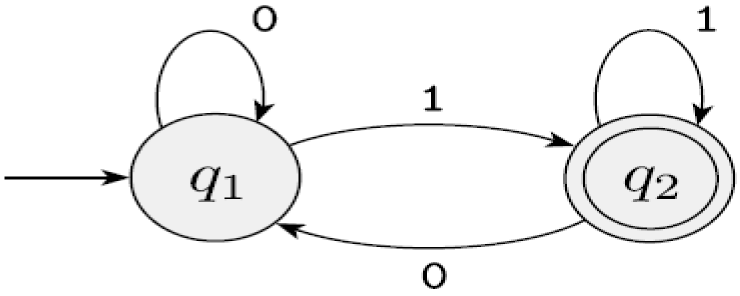
Autômatos Finitos Não Determinísticos

Profa. Ariane Machado Lima

Aula passada

Autômatos Finitos Determinísticos (AFD)

- Dado um estado atual e um símbolo de entrada sabemos exatamente para onde ir (está determinado)



Um *autômato finito* é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

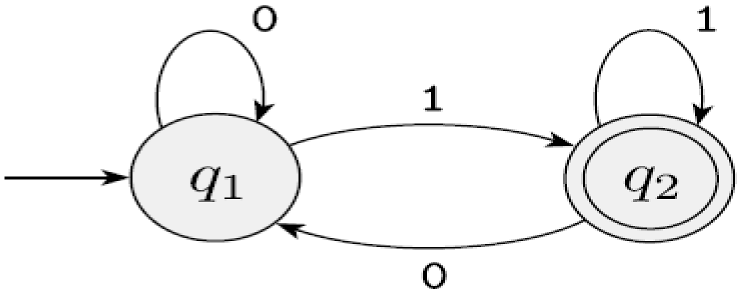
1. Q é um conjunto finito conhecido como os *estados*,
2. Σ é um conjunto finito chamado o *alfabeto*,
3. $\delta \colon Q \times \Sigma \longrightarrow Q$ é a *função de transição*,[1]
4. $q_0 \in Q$ é o *estado inicial*, e
5. $F \subseteq Q$ é o *conjunto de estados de aceitação*.[2]

Note que para um AFD deve haver, saindo de cada estado, uma aresta para CADA símbolo do alfabeto

Autômatos Finitos Determinísticos (AFD)

- Dado um estado atual e um símbolo de entrada sabemos exatamente para onde ir (está determinado)

Por isso a tabela que define o AFD deve estar totalmente preenchida !!!



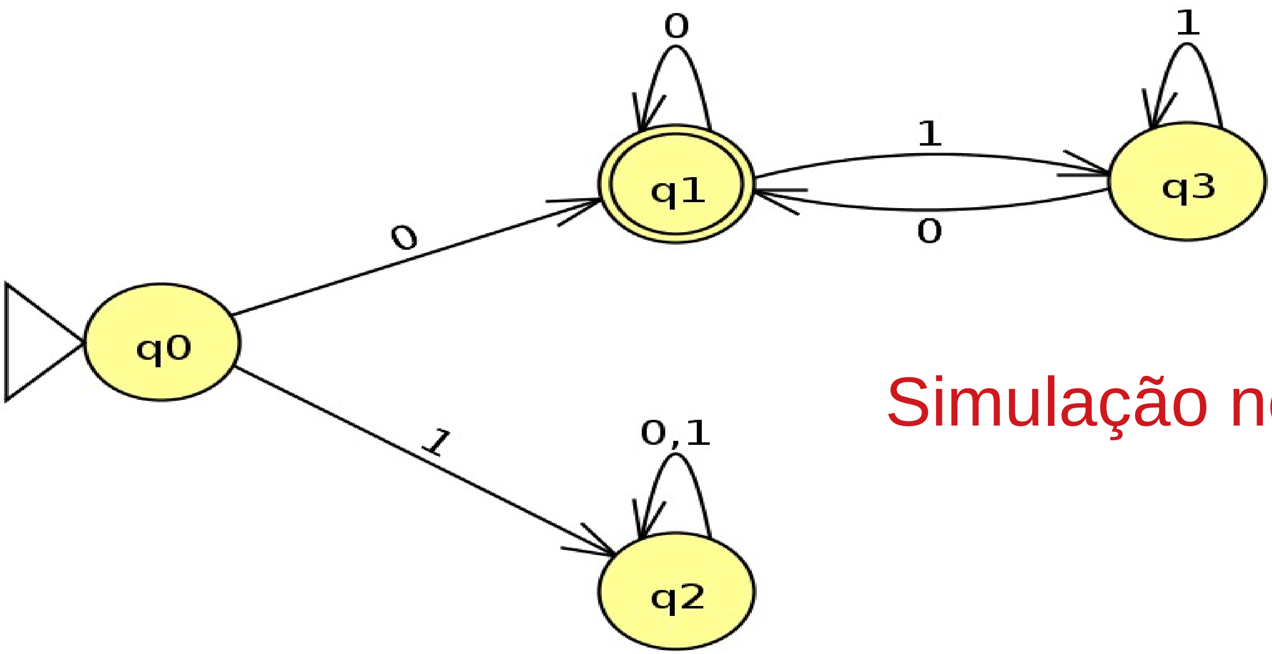
		0	1
→	**q1**	q1	q2
←	**q2**	q1	q2

Na notação tabular, como se pode perceber, o estado inicial é indicado através do símbolo "→", ao passo que os estados finais são indicados por "←". O símbolo "↔" indica um estado que seja simultaneamente inicial e final.

Exercício 1 - resposta

...que comecem e terminem com zero, com tamanho pelo menos 1

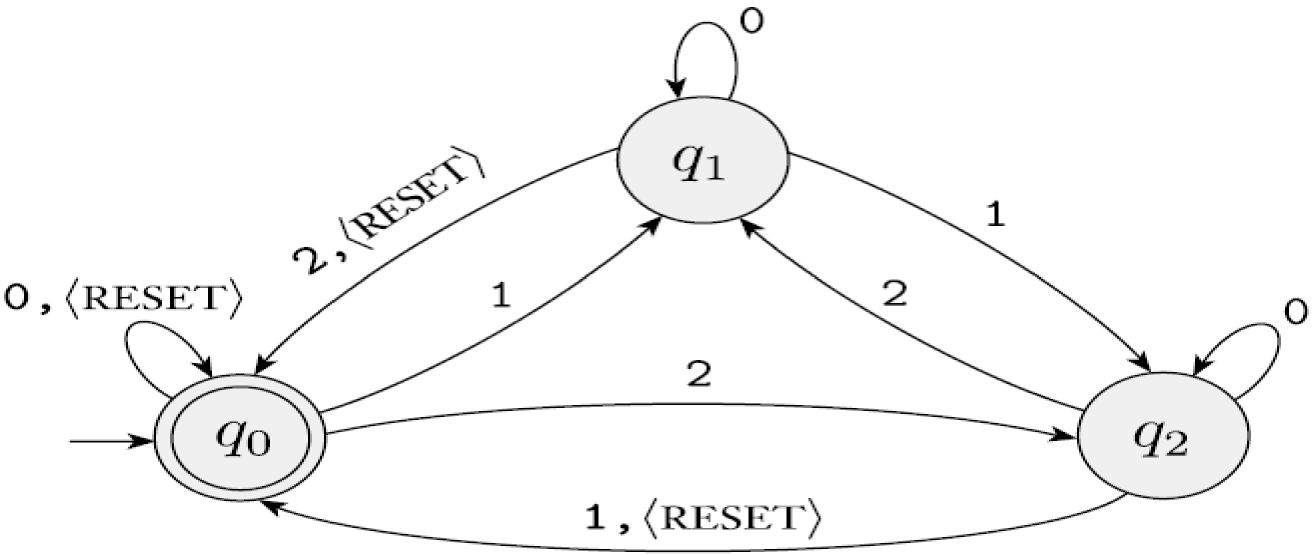
0, 00, 010, 000000, 0101110, ...



Simulação no JFlap

Exercício

- Projete um AFD (diagrama de estados) que, dado Σ = {0,1,2,<RESET>}, aceita a cadeia de entrada se a soma dos números for igual a 0 módulo 3 (ou seja, se a soma for um múltiplo de 3). <RESET> zera o contador. Cadeia vazia também é aceita.



Definição formal de computação

Seja $M = (Q, \Sigma, \delta, q_0, F)$ um autômato finito e suponha que $w = w_1 w_2 \cdots w_n$ seja uma cadeia onde cada w_i é um membro do alfabeto Σ. Então M ***aceita*** w se existe uma seqüência de estados $r_0, r_1, \ldots, r_n$ em Q com três condições:

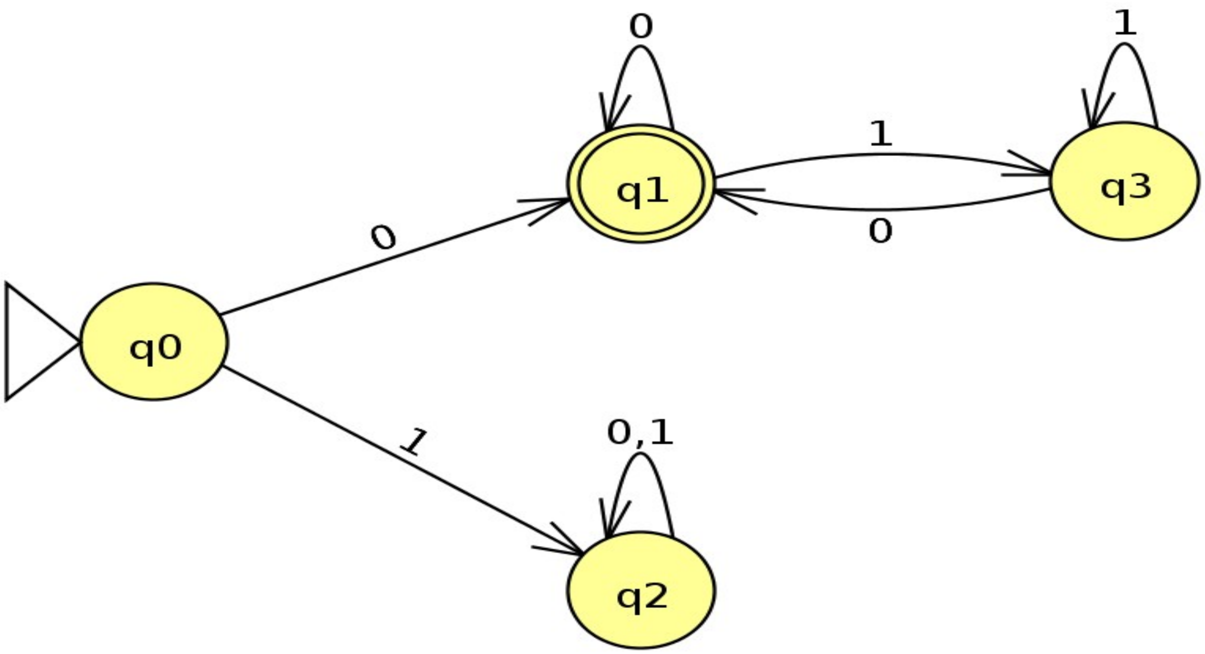
1. $r_0 = q_0$,

2. $\delta(r_i, w_{i+1}) = r_{i+1}$, para $i = 0, \ldots, n-1$, e

3. $r_n \in F$.

Qual a complexidade (tempo) de análise de uma cadeia por um AFD?

- **O(n)** – n sendo o tamanho da cadeia de entrada



Linguagem Regular

- **Definição:** Uma linguagem é chamada **linguagem regular** se algum autômato finito determinístico a reconhece

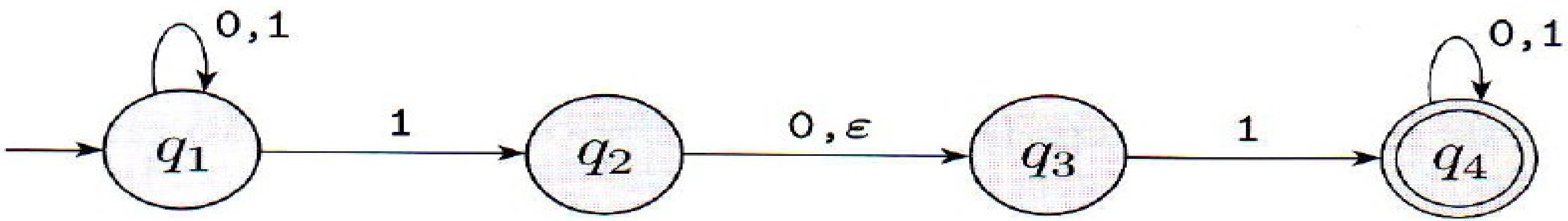
Autômato Mínimo Único

Toda linguagem regular possui um AFD mínimo (em termos de número de estados) **único**

Algoritmo de minimização de AFDs

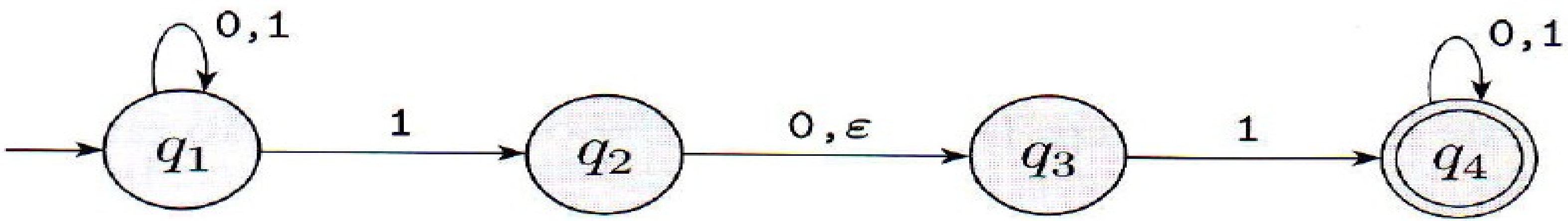
Aula de Hoje

Autômatos Finitos
Não Determinísticos (AFN)



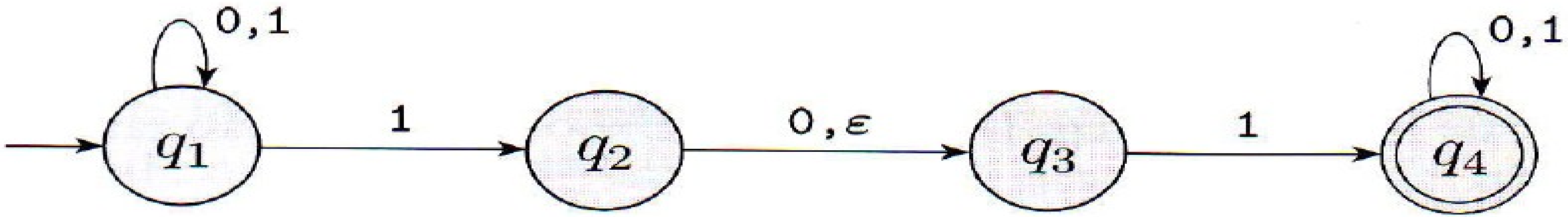
O que neste autômato fere a definição de AFD?

Autômatos Finitos
Não Determinísticos (AFN)



- Um estado pode ter 0 ou mais transições (setas saindo) para cada símbolo de Σ
- Um estado pode ter setas rotuladas por ε ou λ (cadeia vazia)

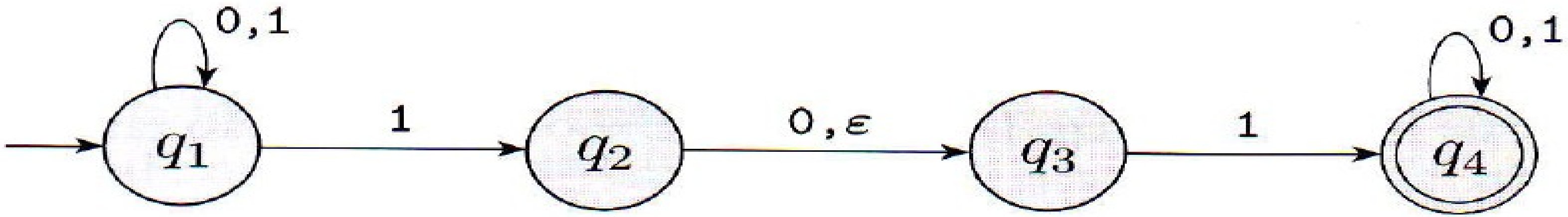
Autômatos Finitos
Não Determinísticos (AFN)



- Um estado pode ter 0 ou mais transições (setas saindo) para cada símbolo de Σ
- Um estado pode ter setas rotuladas por ε ou λ (cadeia vazia)

Como ficaria a função de transição neste caso???

Autômatos Finitos
Não Determinísticos (AFN)



Um ***autômato finito não-determinístico*** é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

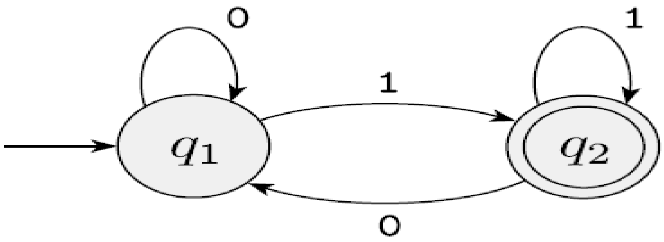
1. Q é um conjunto finito de estados,
2. Σ é um alfabeto finito,
3. $\delta : Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ é a função de transição,
4. $q_0 \in Q$ é o estado inicial, e
5. $F \subseteq Q$ é o conjunto de estados de aceitação.

$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

Conjunto potência

Aulas passadas

AFD

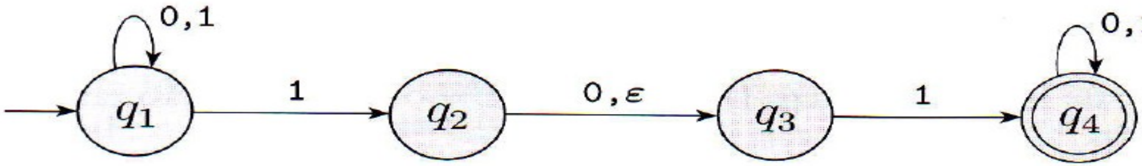


Para cada par (estado atual, próximo símbolo) está DETERMINADO qual é o próximo estado

Um *autômato finito* é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

1. Q é um conjunto finito conhecido como os ***estados***,
2. Σ é um conjunto finito chamado o ***alfabeto***,
3. $\delta : Q \times \Sigma \longrightarrow Q$ é a ***função de transição***,[1]
4. $q_0 \in Q$ é o ***estado inicial***, e
5. $F \subseteq Q$ é o ***conjunto de estados de aceitação***.[2]

AFN

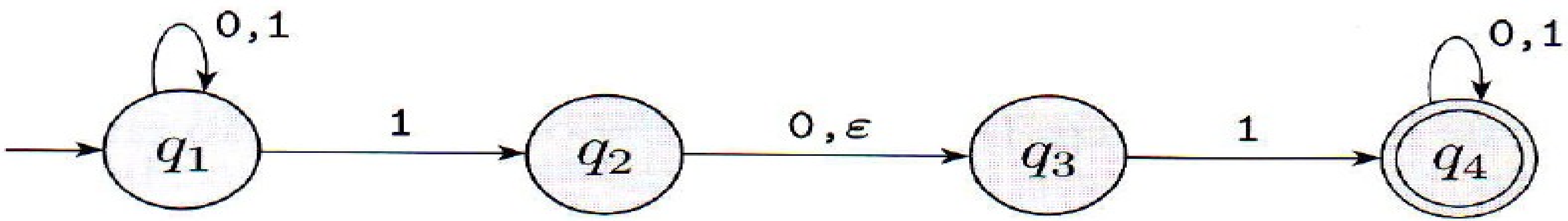


Um *autômato finito não-determinístico* é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, onde

1. Q é um conjunto finito de estados,
2. Σ é um alfabeto finito,
3. $\delta : Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ é a função de transição,
4. $q_0 \in Q$ é o estado inicial, e
5. $F \subseteq Q$ é o conjunto de estados de aceitação.

Para cada par (estado atual, próximo símbolo – incluindo ε) há um conjunto de estados possíveis

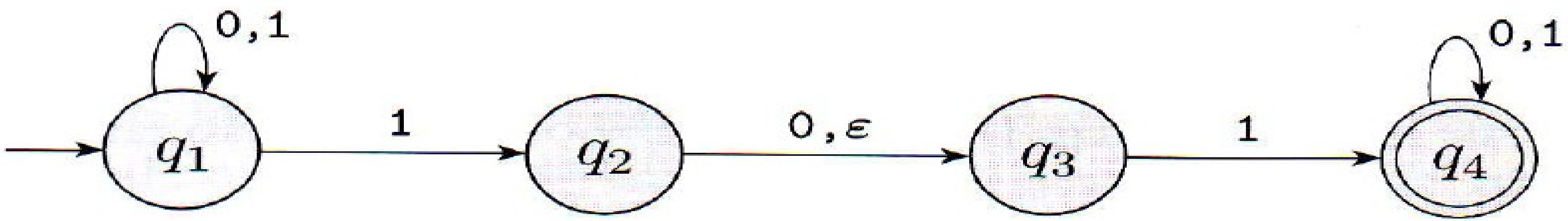
Autômatos Finitos Não Determinísticos (AFN)



- δ:

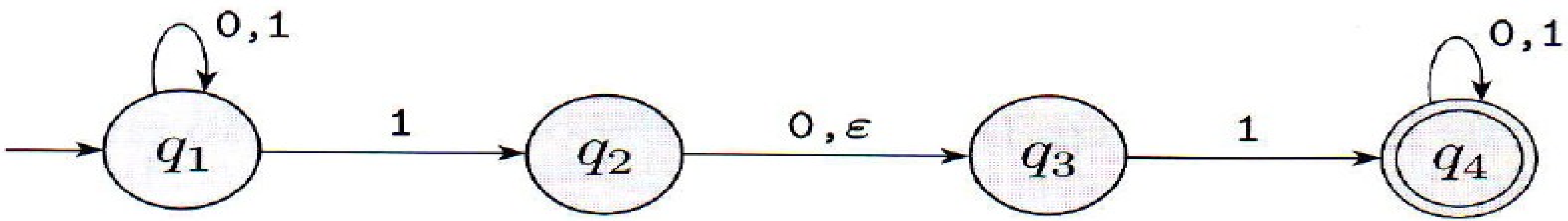
	Q \ Σ	0	1	ε
→	q1	{q1}	{q1,q2}	∅
	q2	{q3}	∅	{q3}
	q3	∅	{q4}	∅
←	q4	{q4}	{q4}	∅

Autômatos Finitos Não Determinísticos (AFN)



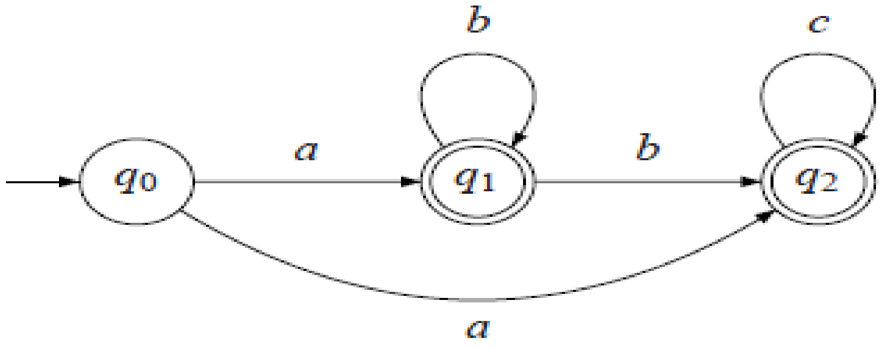
- Que linguagem este AFN reconhece?

Autômatos Finitos
Não Determinísticos (AFN)



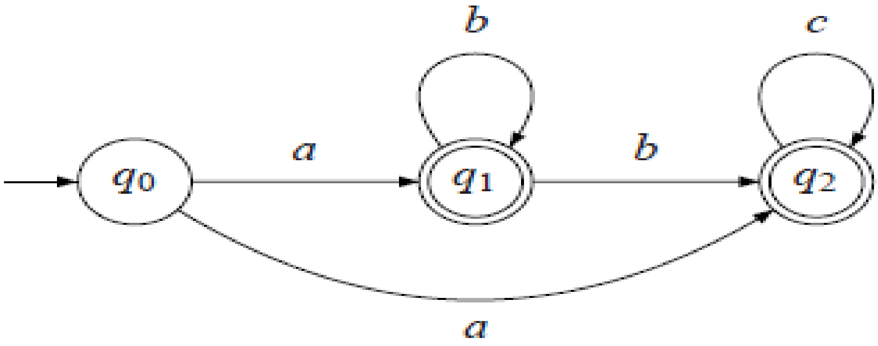
- Que linguagem este AFN reconhece?
- Sequências binárias que contenham 11 ou 101

Exercício: como seria a representação tabular deste AFN? E que linguagem ele reconhece?



Na notação tabular, como se pode perceber, o estado inicial é indicado através do símbolo “→”, ao passo que os estados finais são indicados por “←”. O símbolo “↔” indica um estado que seja simultaneamente inicial e final.

Representação tabular deste AFN

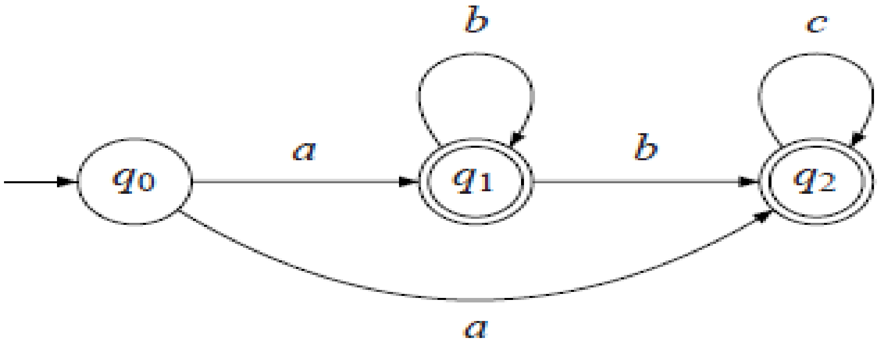


	δ	a	b	c	ε
$\rightarrow$	q_0	$\{q_1, q_2\}$	Ø	Ø	Ø
$\leftarrow$	q_1	Ø	$\{q_1, q_2\}$	Ø	Ø
$\leftarrow$	q_2	Ø	Ø	$\{q_2\}$	Ø

Na notação tabular, como se pode perceber, o estado inicial é indicado através do símbolo "$\rightarrow$", ao passo que os estados finais são indicados por "$\leftarrow$". O símbolo "$\leftrightarrow$" indica um estado que seja simultaneamente inicial e final.

E que linguagem esse AFN descreve?

	δ	a	b	c	ε
$\rightarrow$	q_0	$\{q_1, q_2\}$	Ø	Ø	Ø
$\leftarrow$	q_1	Ø	$\{q_1, q_2\}$	Ø	Ø
$\leftarrow$	q_2	Ø	Ø	$\{q_2\}$	Ø

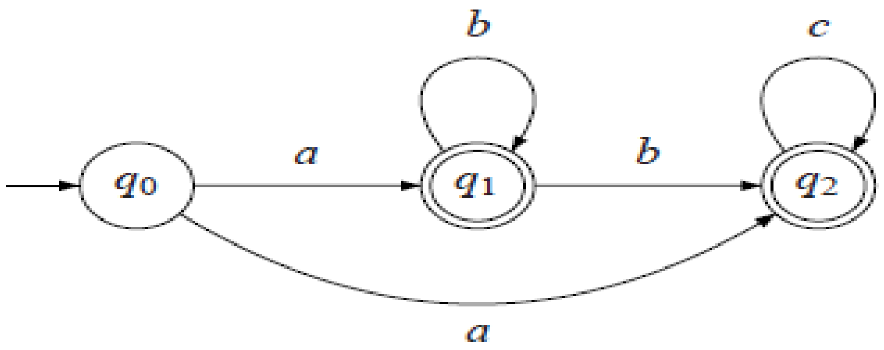


Na notação tabular, como se pode perceber, o estado inicial é indicado através do símbolo "$\rightarrow$", ao passo que os estados finais são indicados por "$\leftarrow$". O símbolo "$\leftrightarrow$" indica um estado que seja simultaneamente inicial e final.

Exercício: que linguagem esse AFN descreve?

(**simplesmente lendo o AFN**:) Sequências que:
Começam com um *a*, seguido por zero ou mais *b*'s (ab*)
ou
Começam com um *a*, seguido por um ou mais *b*'s, seguidos por zero ou mais *c*'s ($ab^{+}c^{*}$)
Ou
Começam com um a, seguido por zero ou mais *c*'s (ac*)

	δ	a	b	c	ε
$\rightarrow$	q_0	$\{q_1, q_2\}$	Ø	Ø	Ø
$\leftarrow$	q_1	Ø	$\{q_1, q_2\}$	Ø	Ø
$\leftarrow$	q_2	Ø	Ø	$\{q_2\}$	Ø



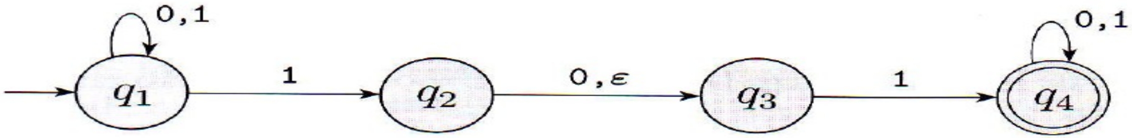
No fundo:

L= {w є {a,b,c}* | w começa com um *a* e depois só podem vir *b*'s ou *c*'s (zero ou mais) desde que os *b*'s venham antes dos *c*'s.

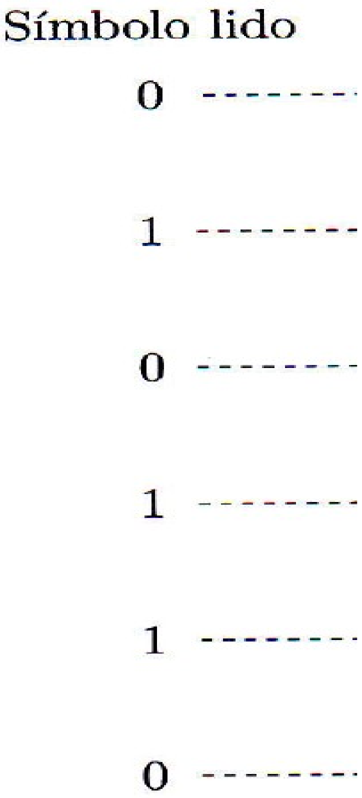
Na notação tabular, como se pode perceber, o estado inicial é indicado através do símbolo "$\rightarrow$", ao passo que os estados finais são indicados por "$\leftarrow$". O símbolo "$\leftrightarrow$" indica um estado que seja simultaneamente inicial e final.

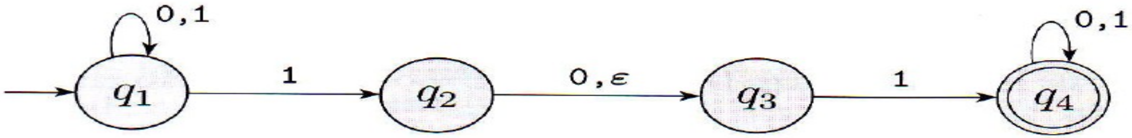
Funcionamento de um AFN

- Sempre que o autômato se depara com um não-determinismo (símbolo repetido ou ε) faz uma cópia de si (um clone), exatamente no ponto onde pausou, e cada cópia segue com uma alternativa, em paralelo, a partir daquele ponto.
- Se alguma cópia aceitar a cadeia, então o AFN aceita a cadeia
- As várias cópias são como várias *threads* ou processos executados em paralelo...

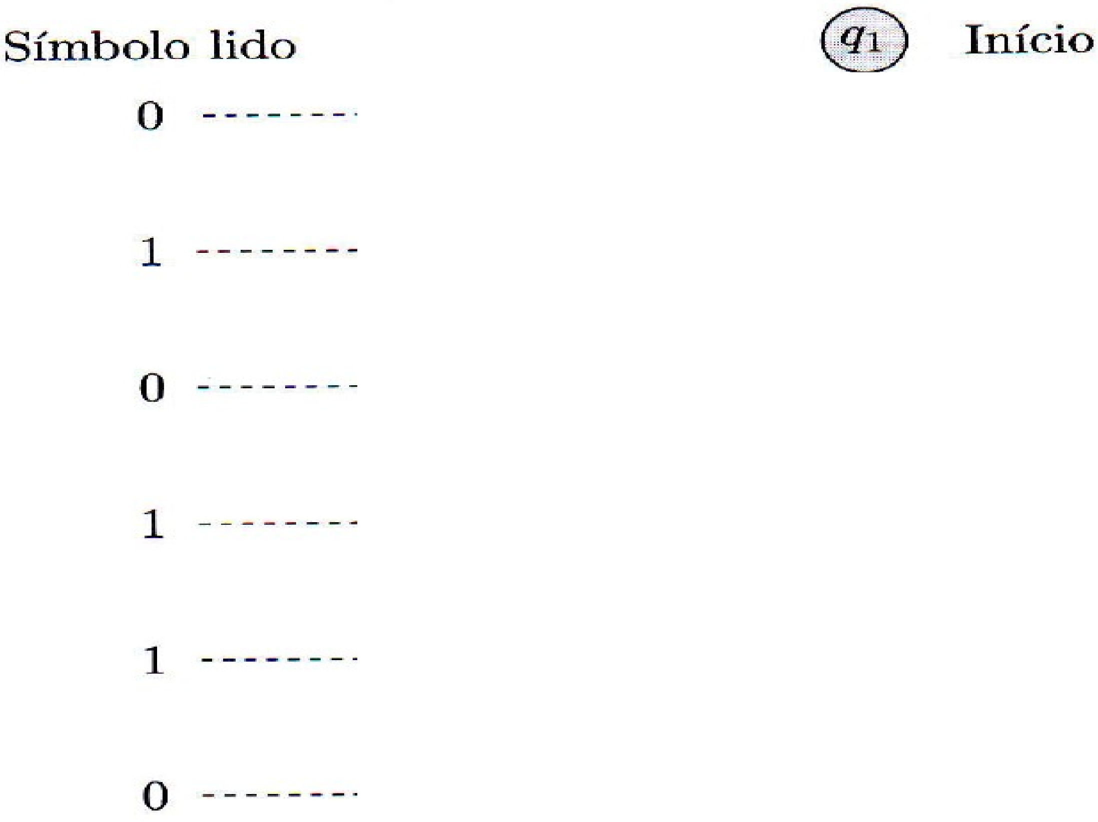


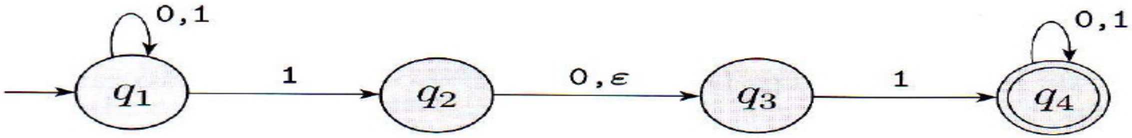
Árvore de computações



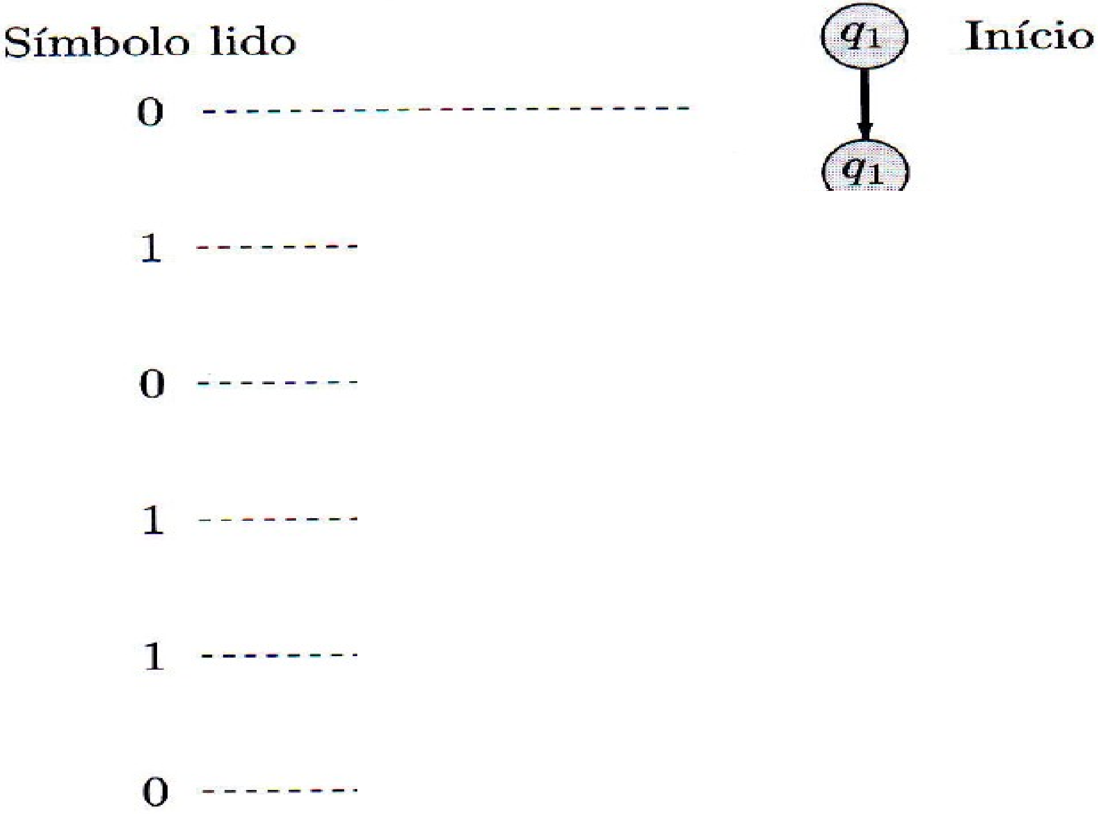


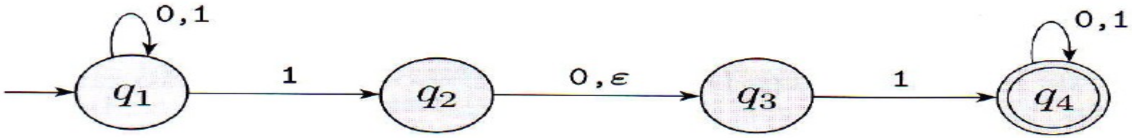
Árvore de computações



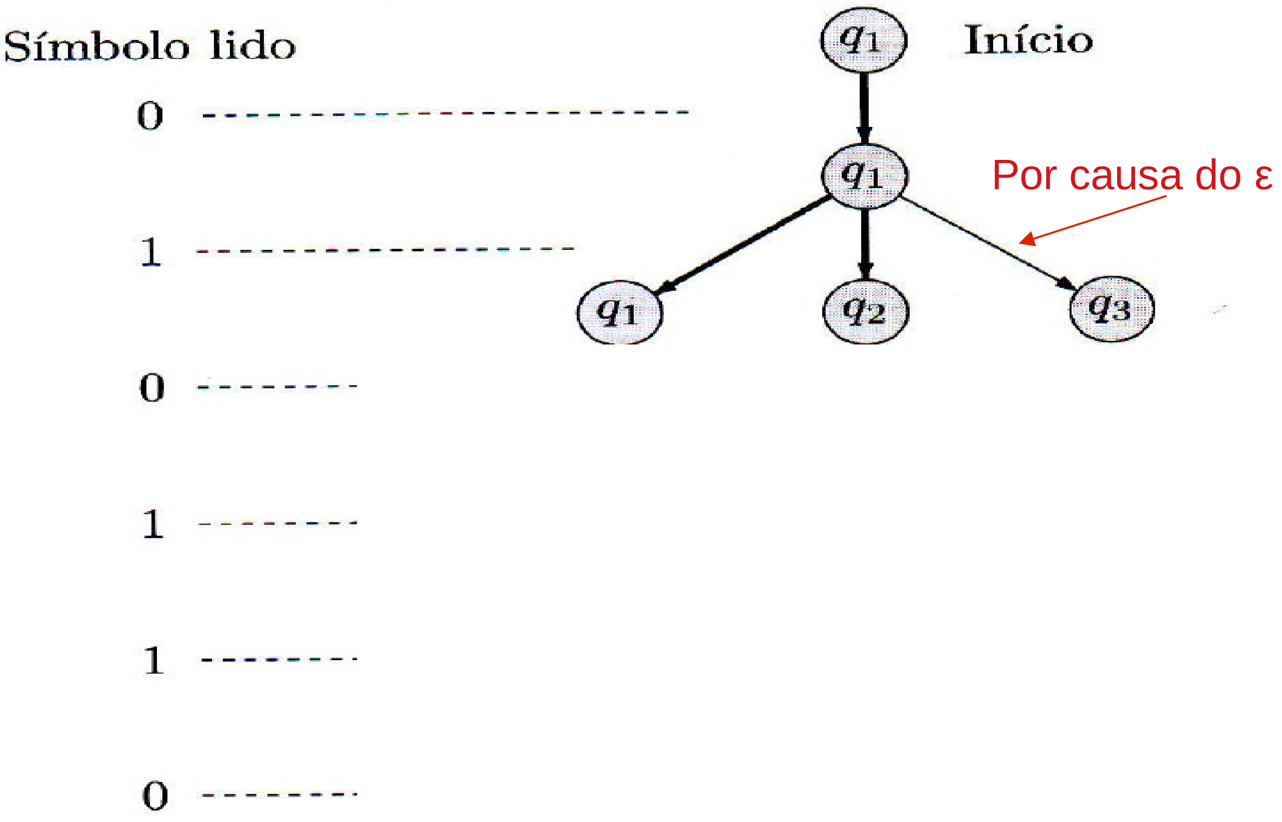


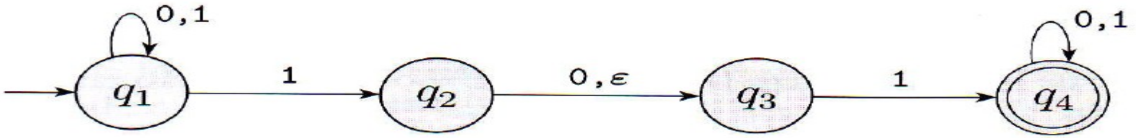
Árvore de computações





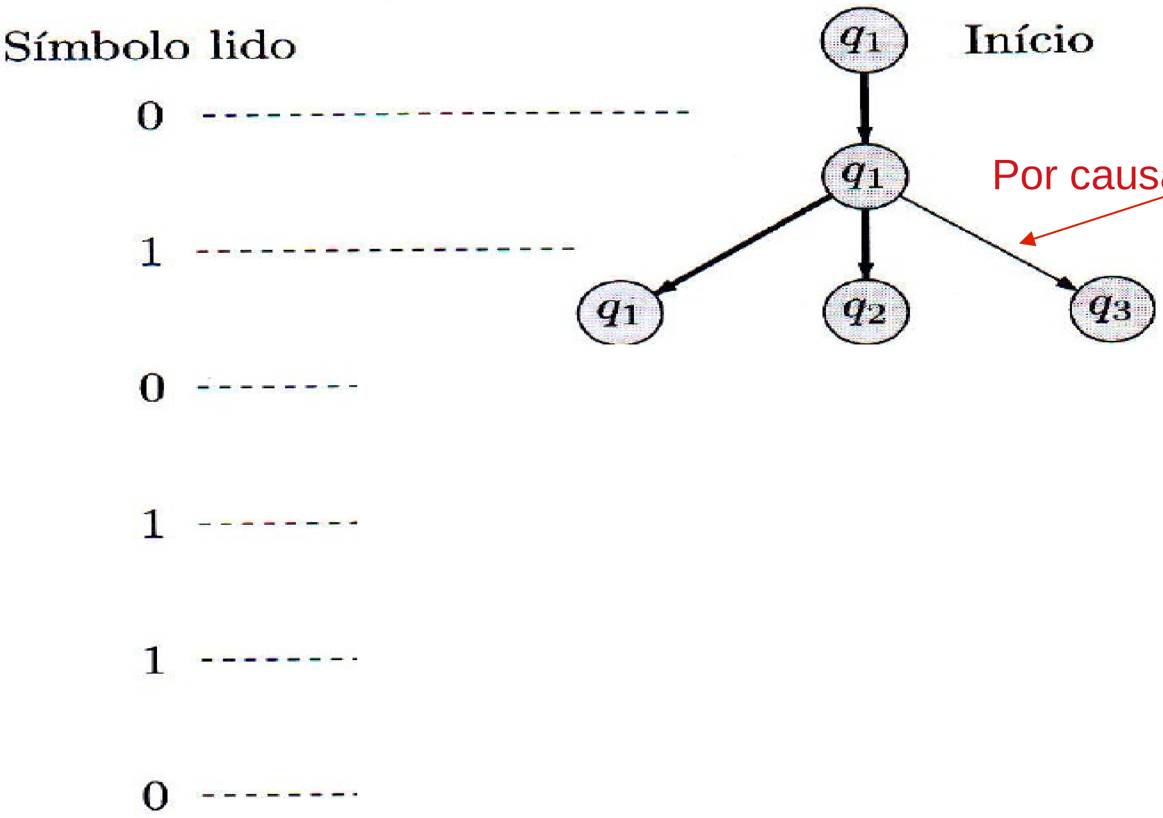
Árvore de computações



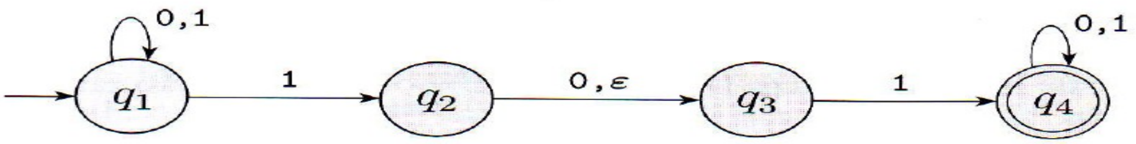


Como q2 tem uma transição no vazio, assim que ele é alcançado a transição é feita sem consumir nenhum símbolo da entrada.

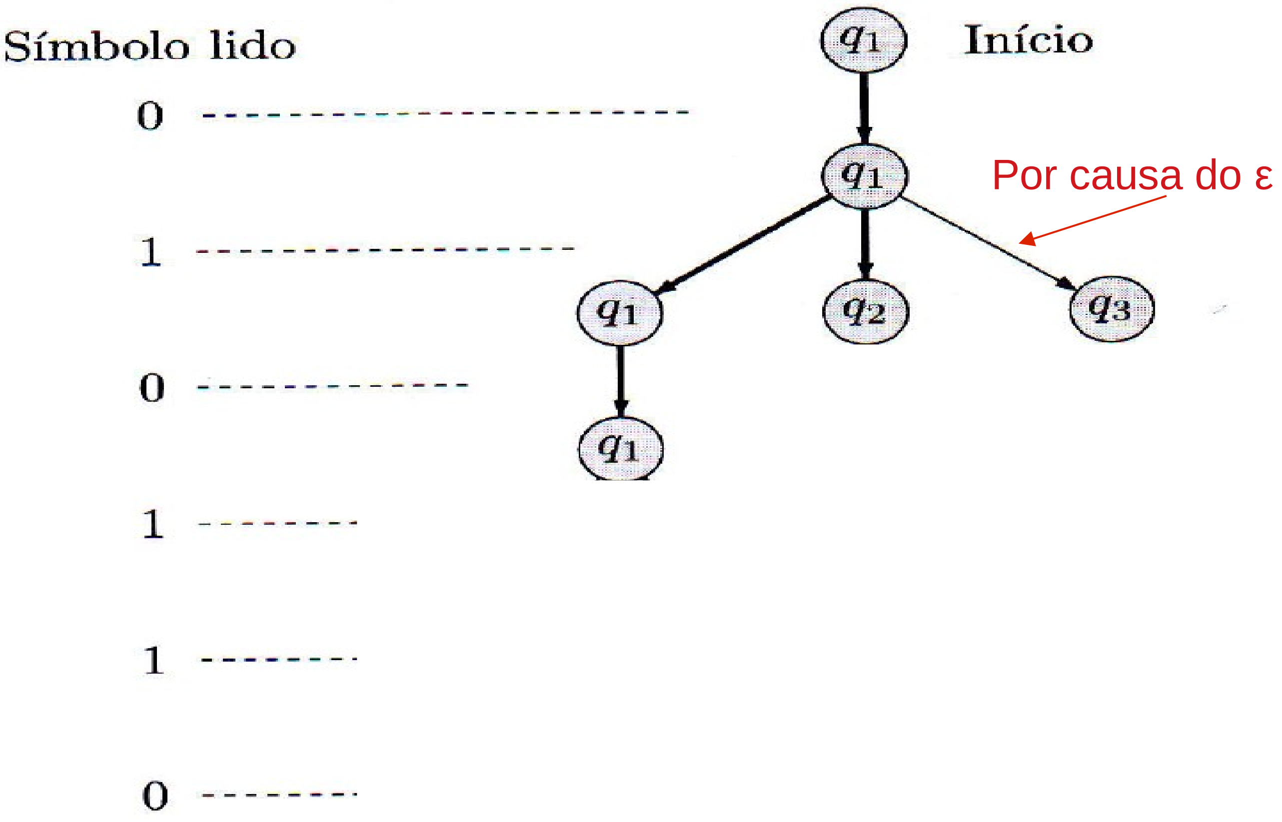
Assim, é possível sair de q1 e ir para q3, passando por q2, consumindo apenas o símbolo “1” da entrada.
SE q3 fosse de aceitação e a cadeia fosse 01, ela teria que ser aceita...

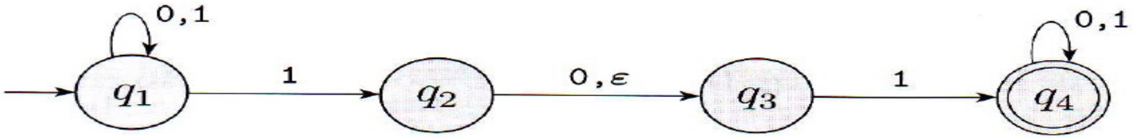


Por causa do ε

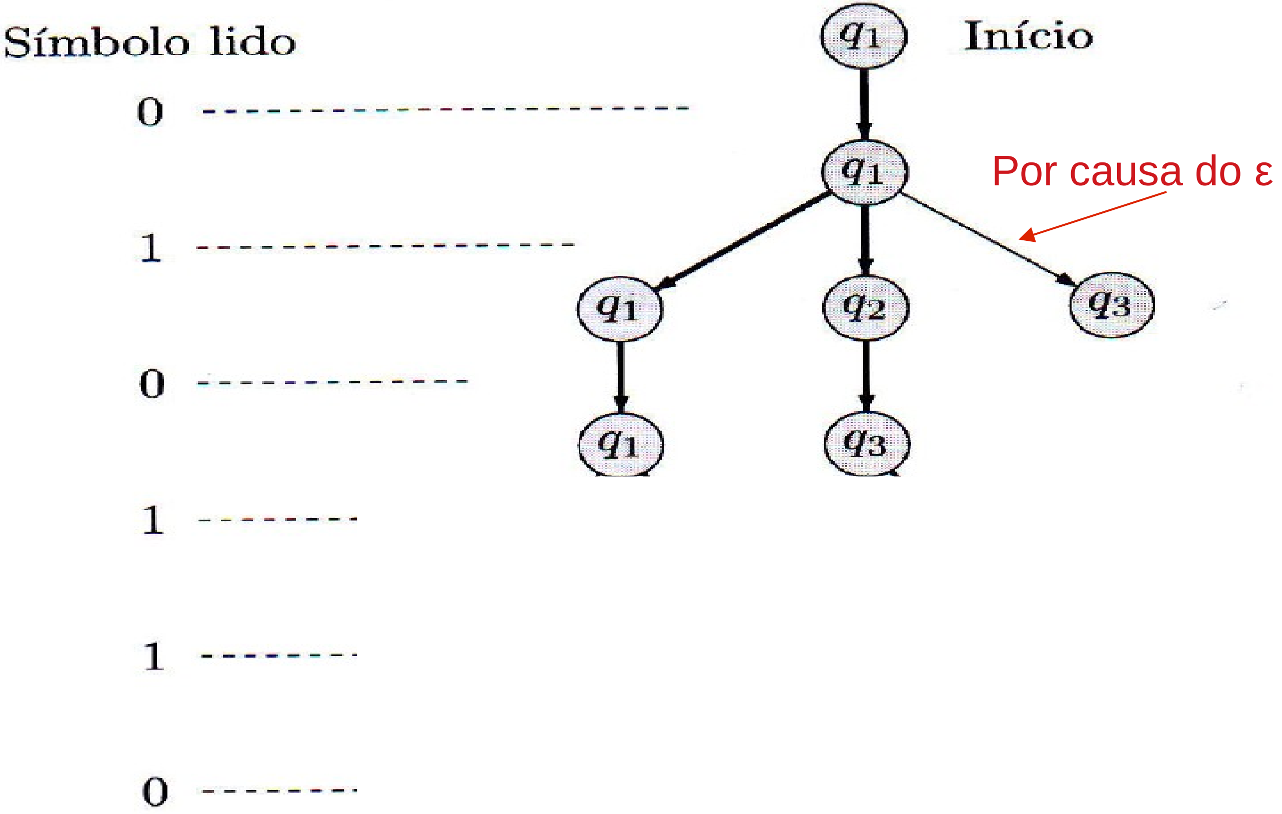


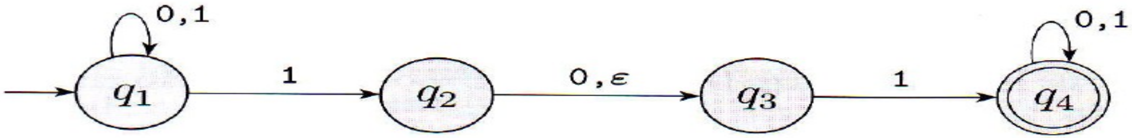
Árvore de computações



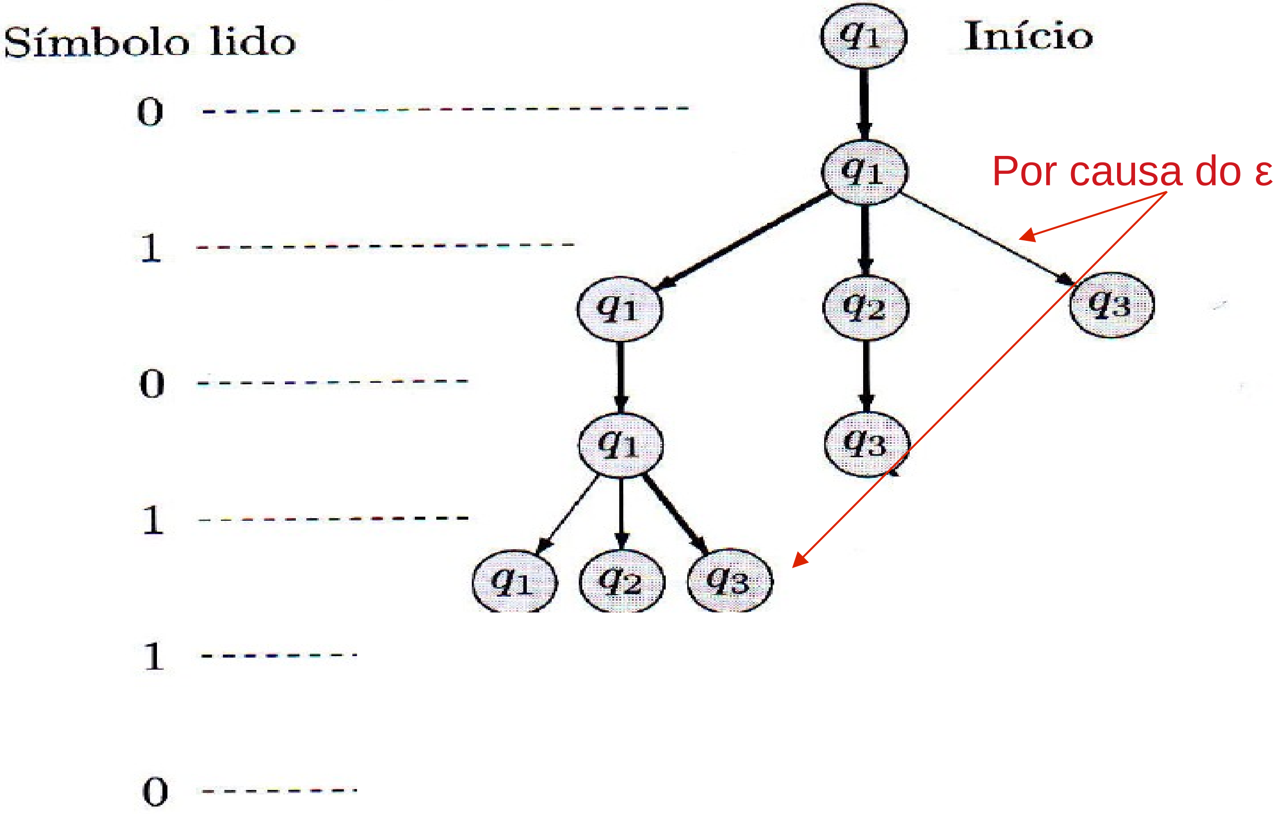


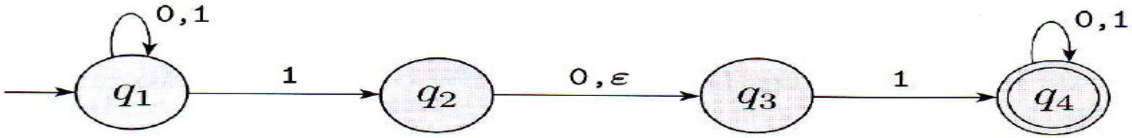
Árvore de computações



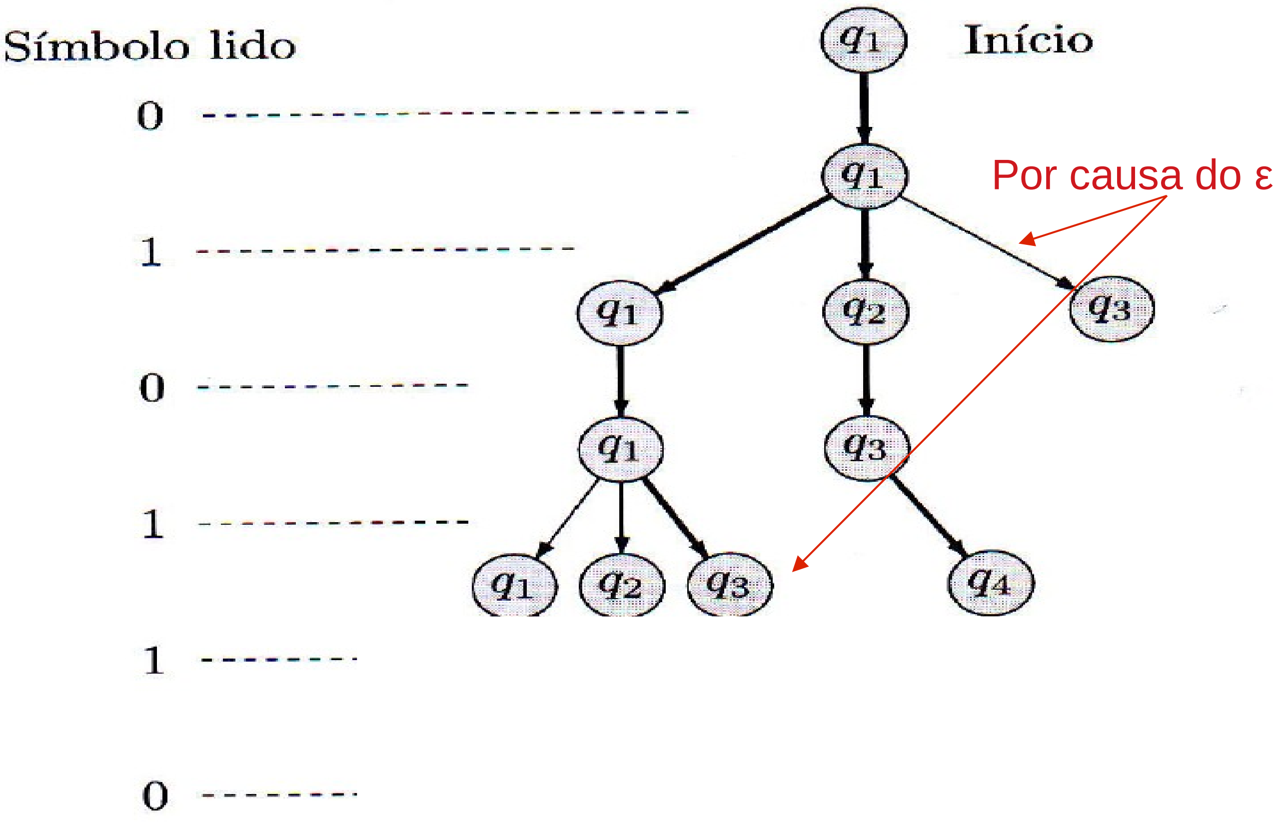


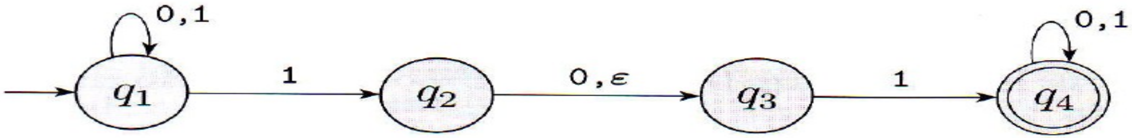
Árvore de computações



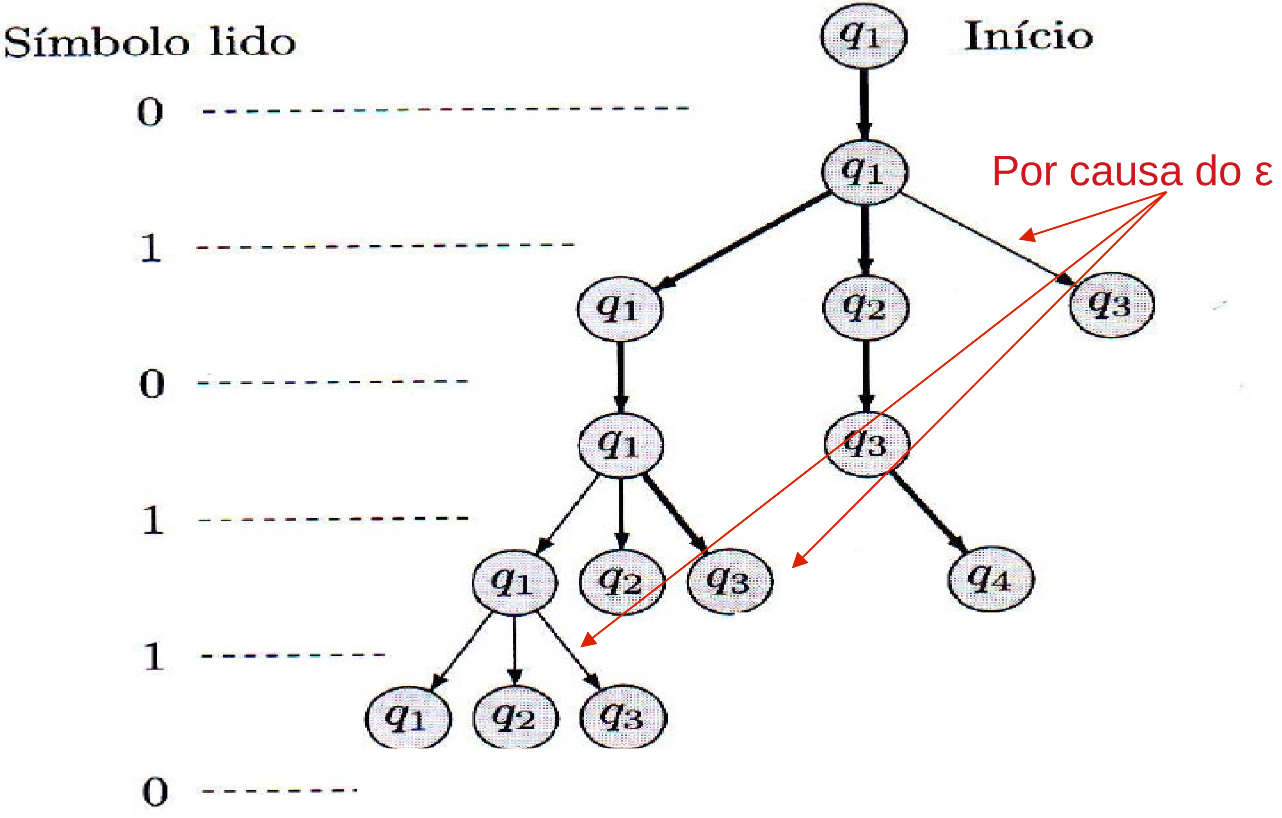


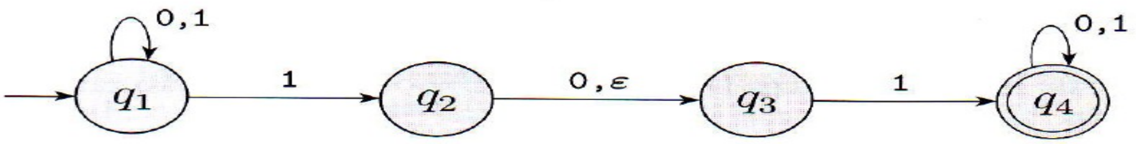
Árvore de computações



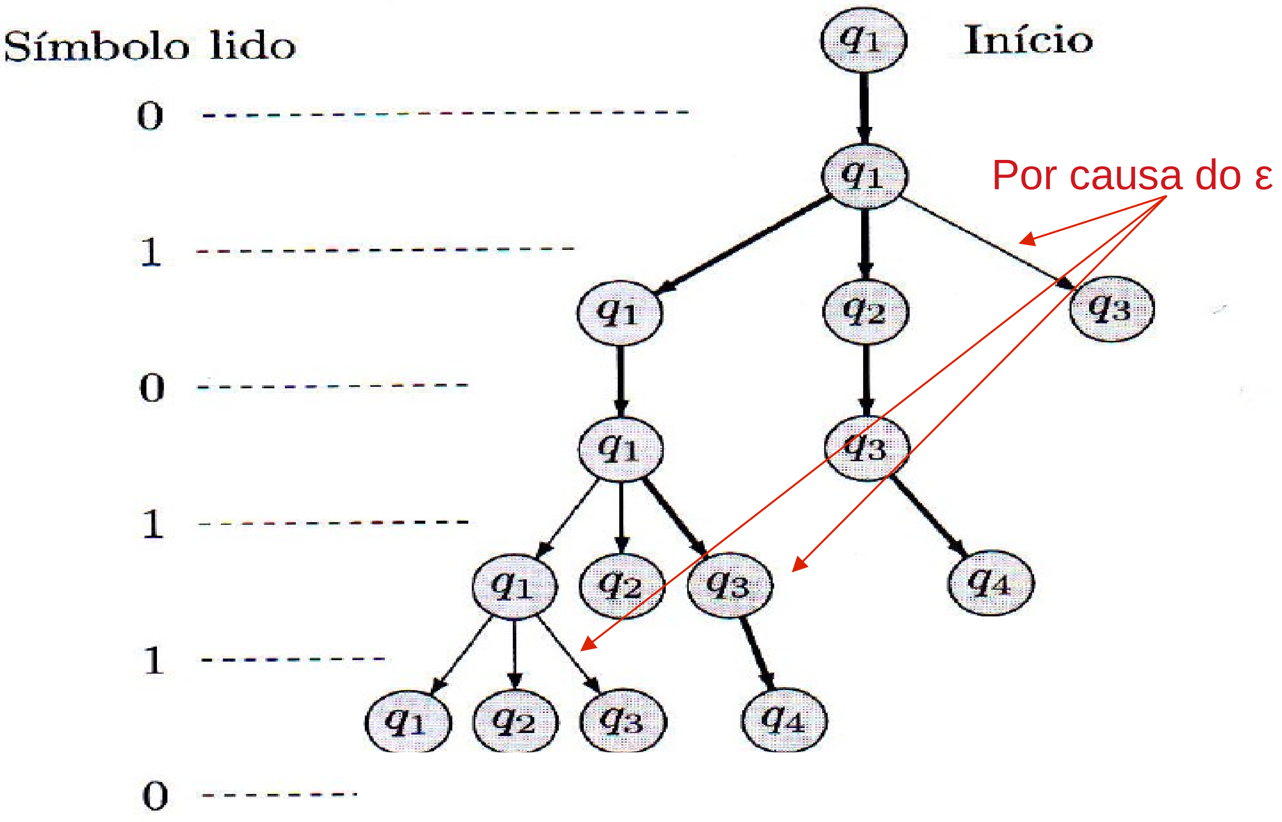


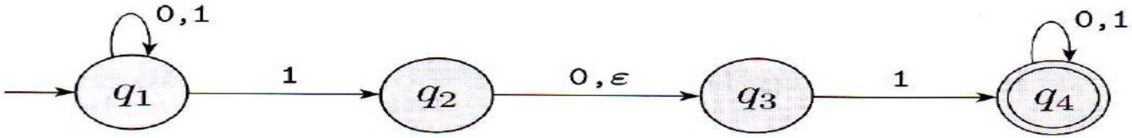
Árvore de computações



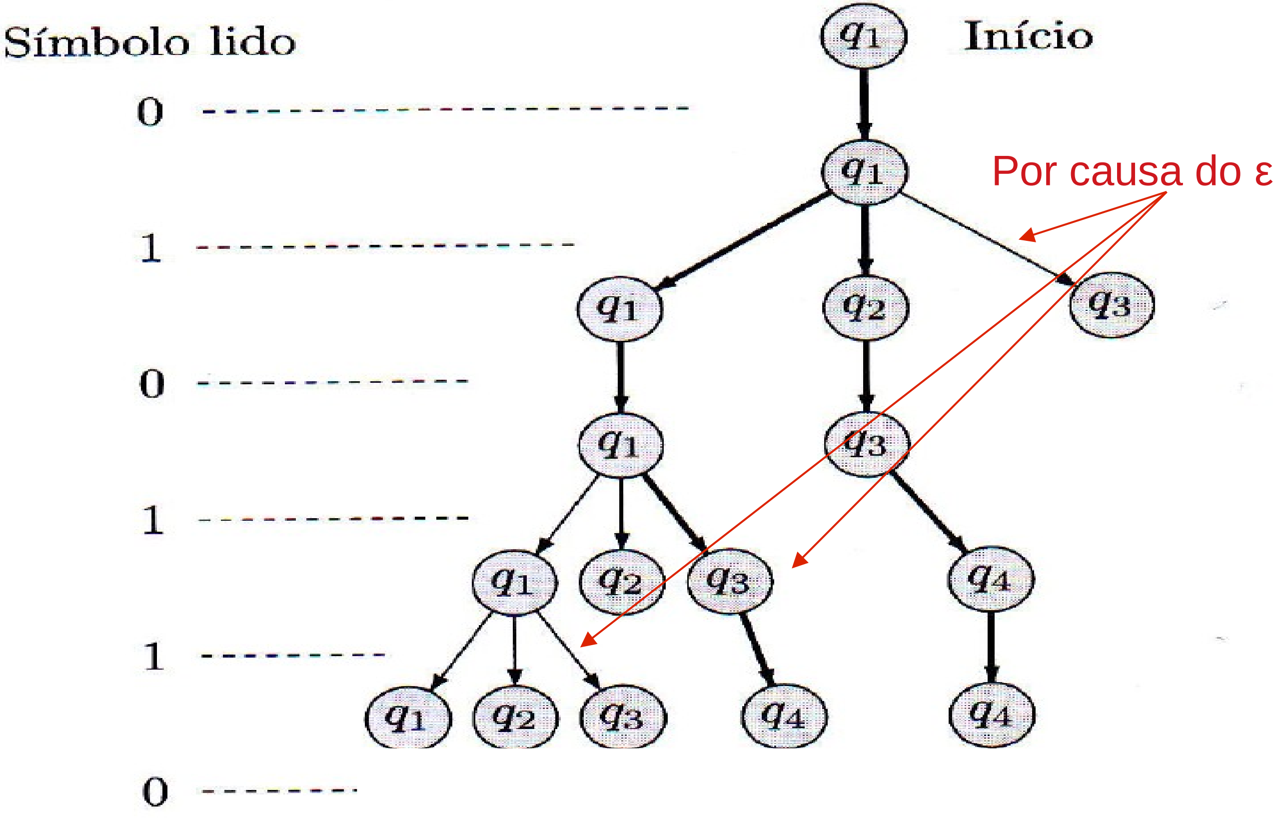


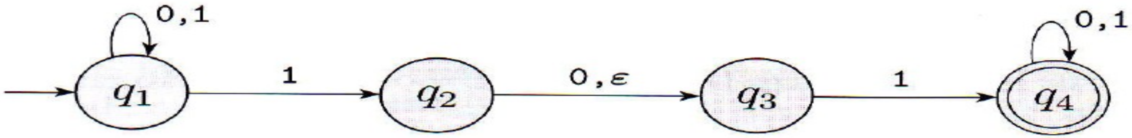
Árvore de computações



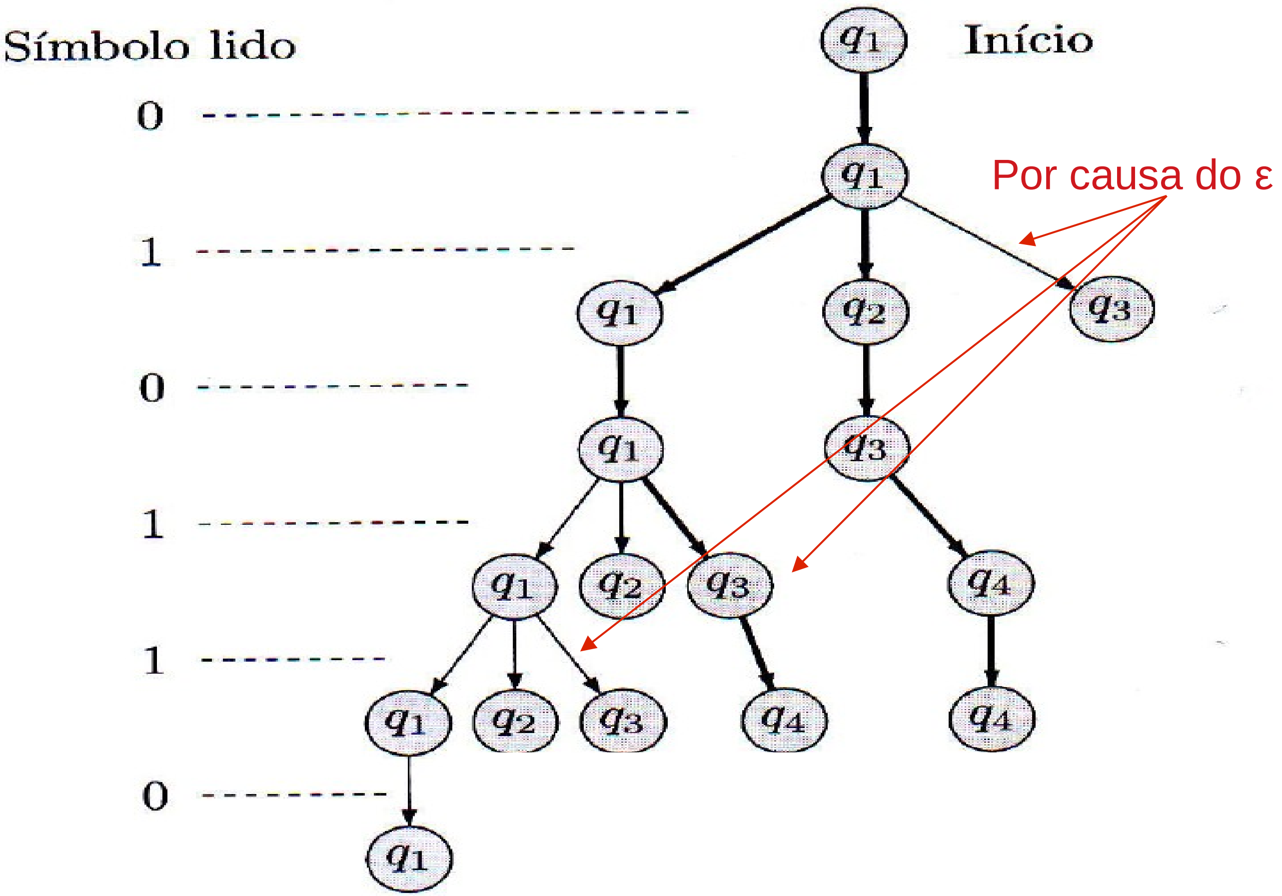


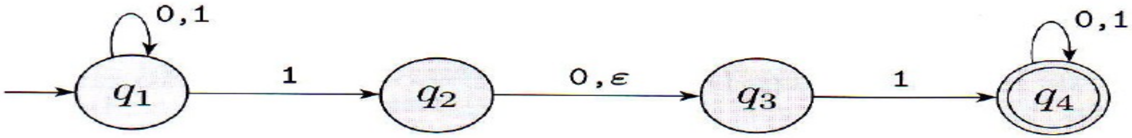
Árvore de computações



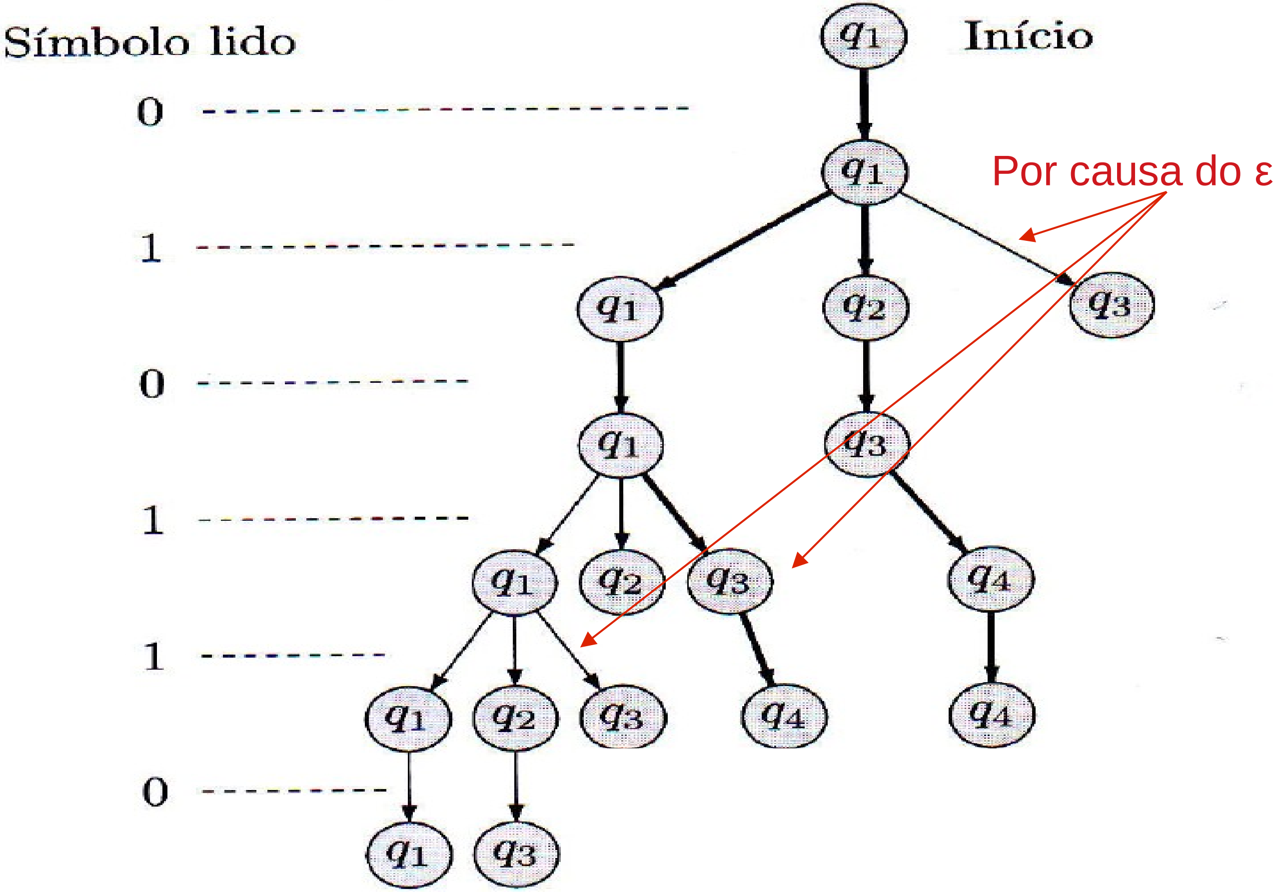


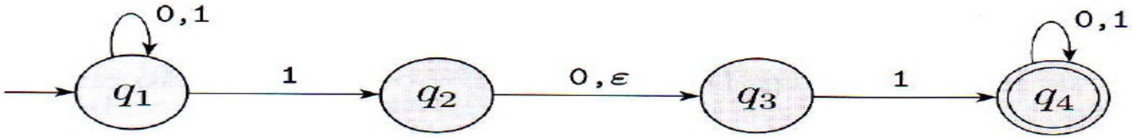
Árvore de computações



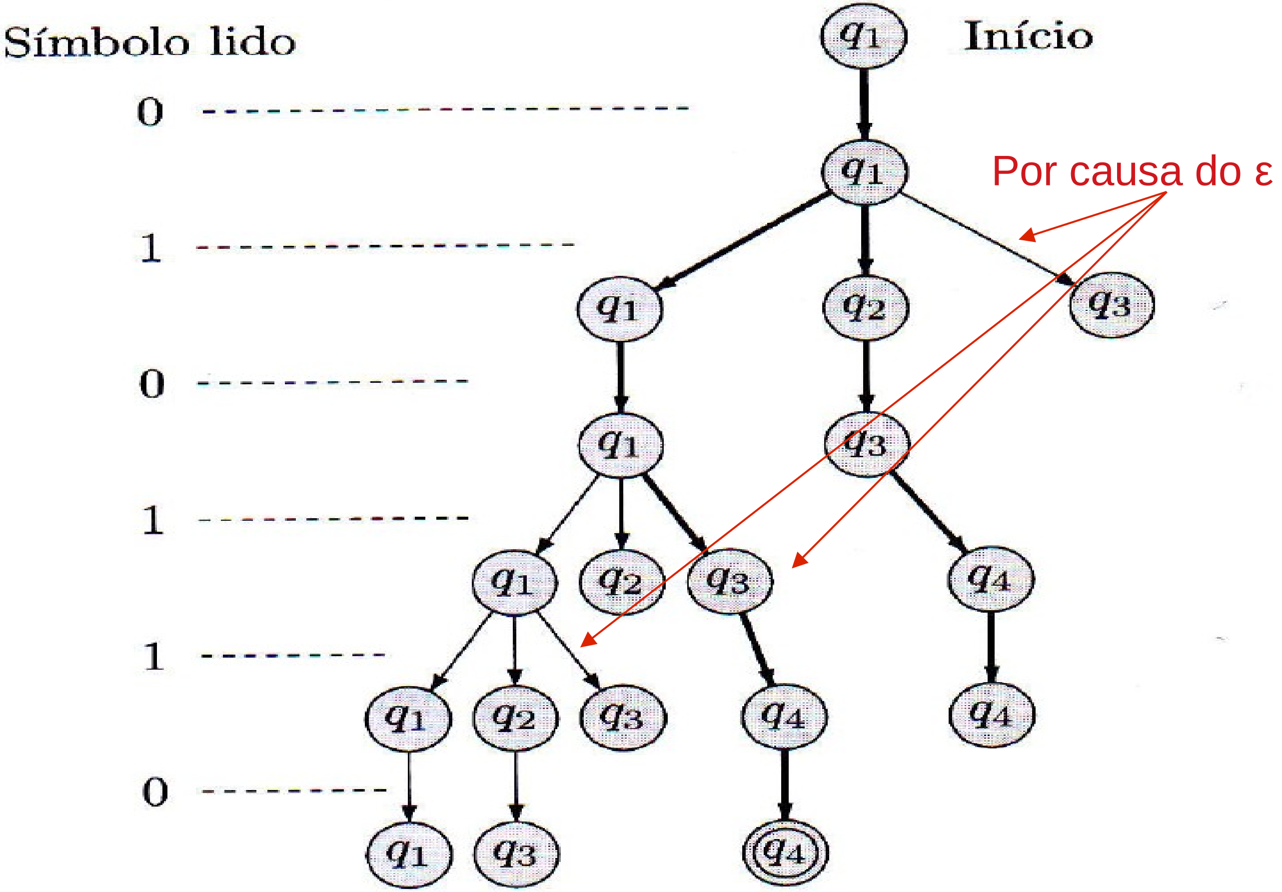


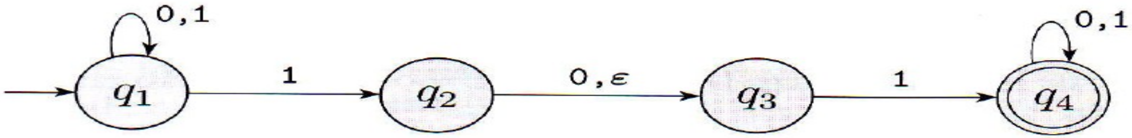
Árvore de computações



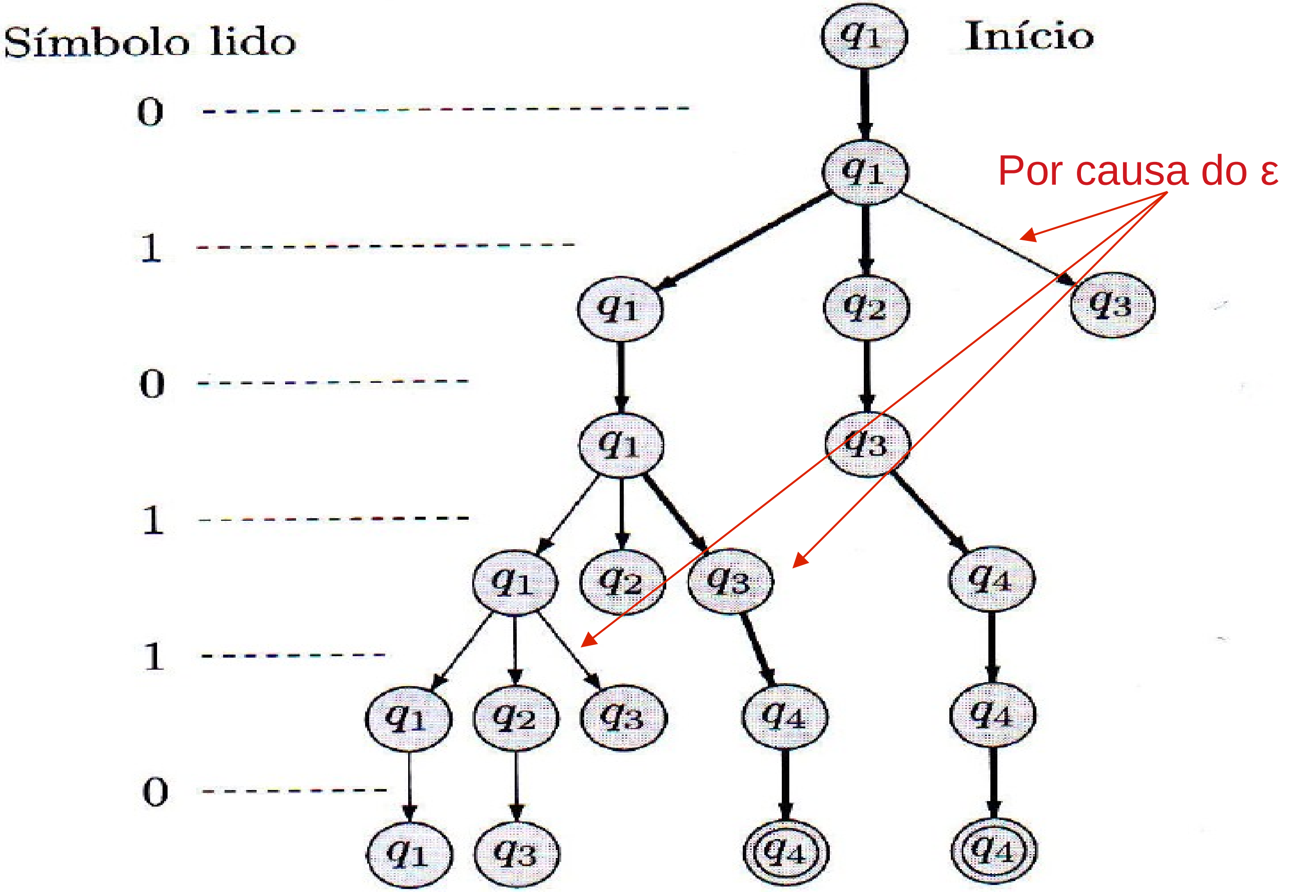


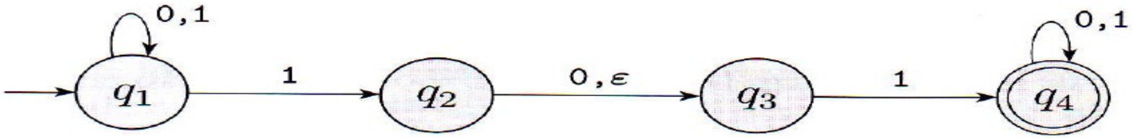
Árvore de computações



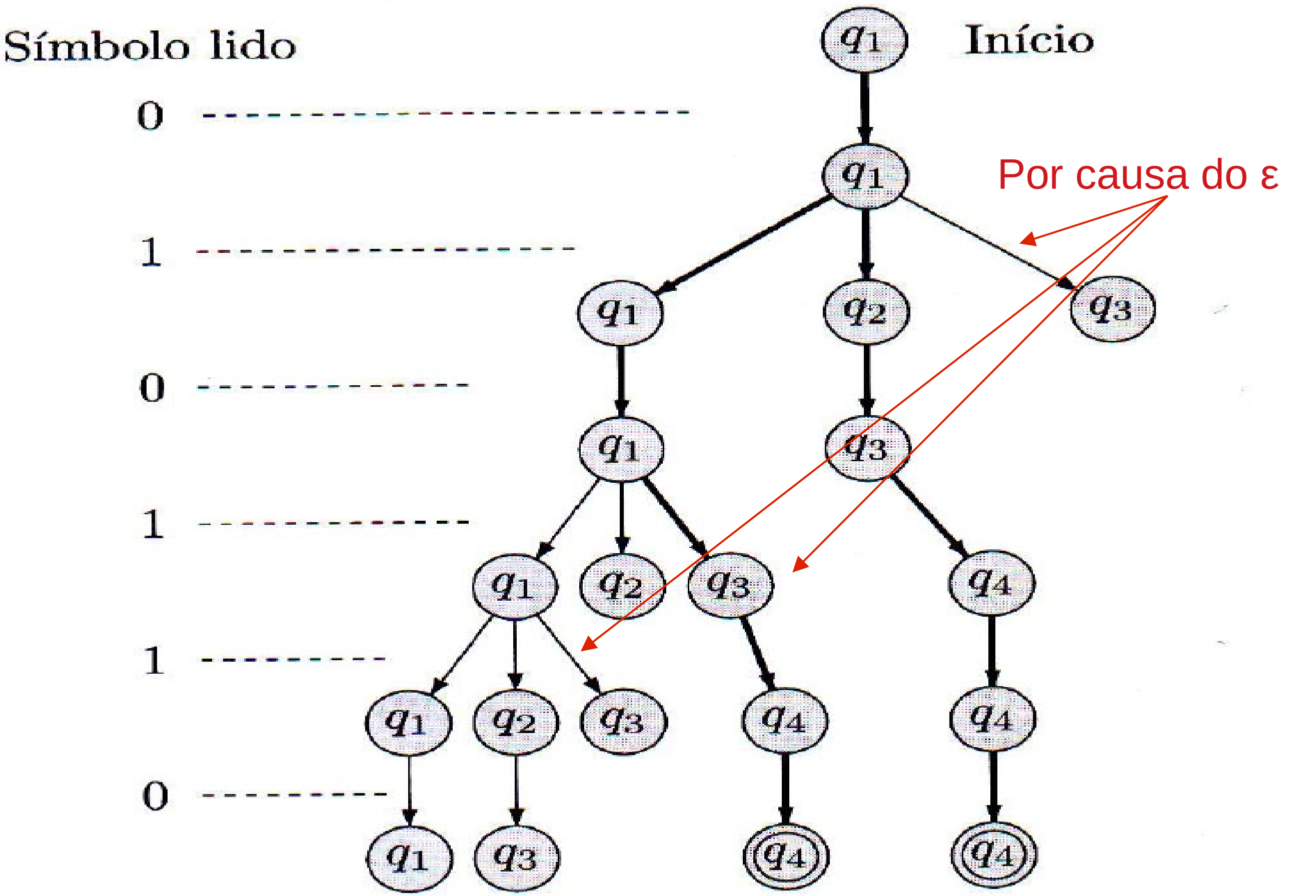


Árvore de computações





Essa cadeia 010110 é aceita por esse AFN ?



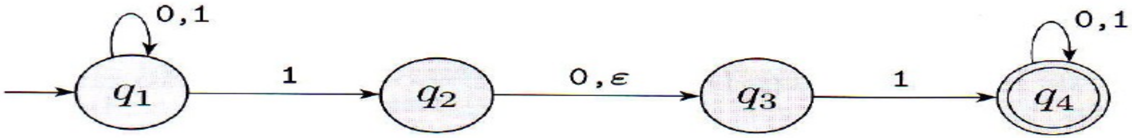
Aceitação e rejeição

Tabela 2: Aceitação e rejeição de cadeias em autômatos finitos

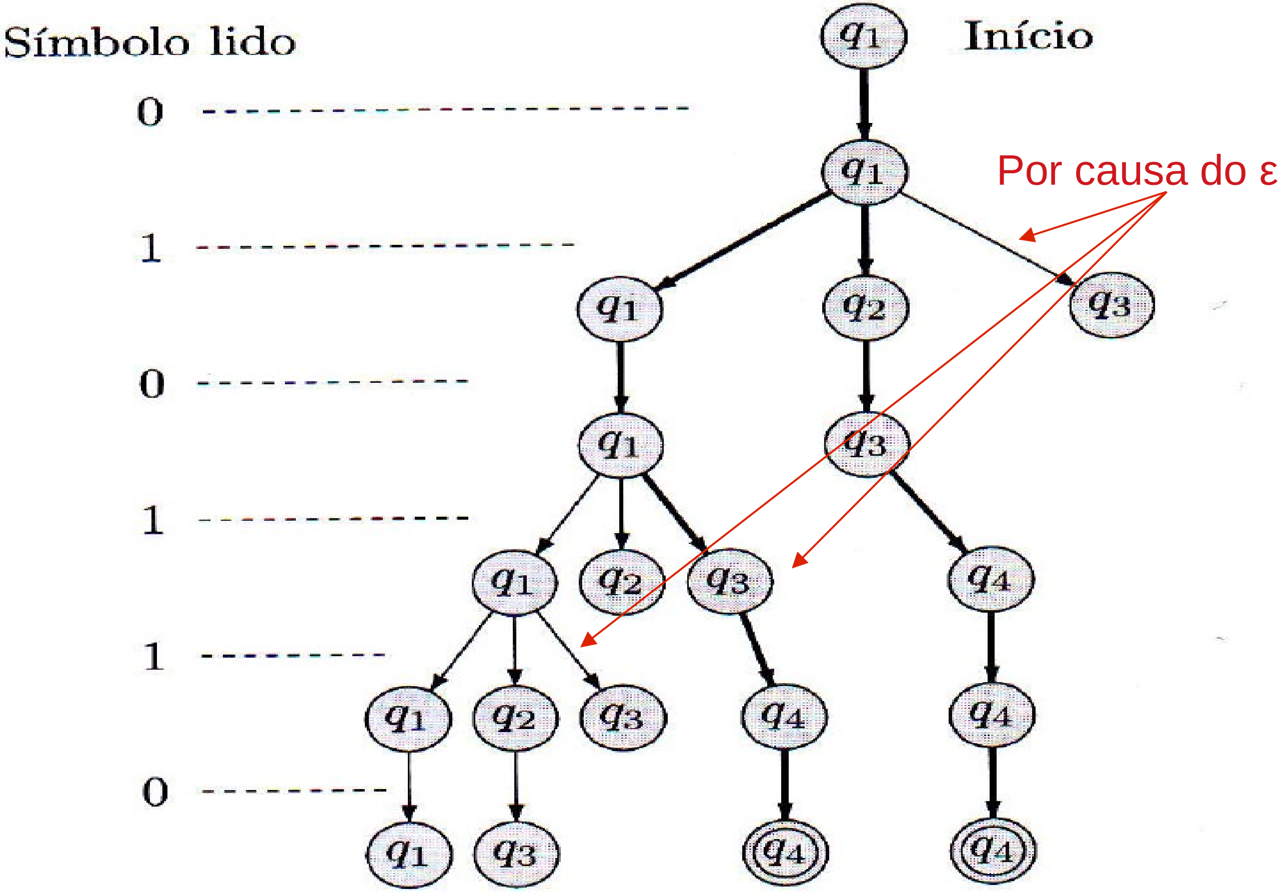
	Dada uma cadeia de entrada, ele:	*Aceita a cadeia de entrada se:*	*Rejeita a cadeia de entrada se:*
Autômato finito determinístico	Executa uma única seqüência de movimentos.	Pára em uma configuração final.	Pára em uma configuração não-final.
Autômato finito não-determinístico	Pode executar várias seqüências distintas de movimentos.	Pára em uma configuração final.	Pára sem conseguir atingir nenhuma configuração final.

Após terminar de ler a cadeia!!!!

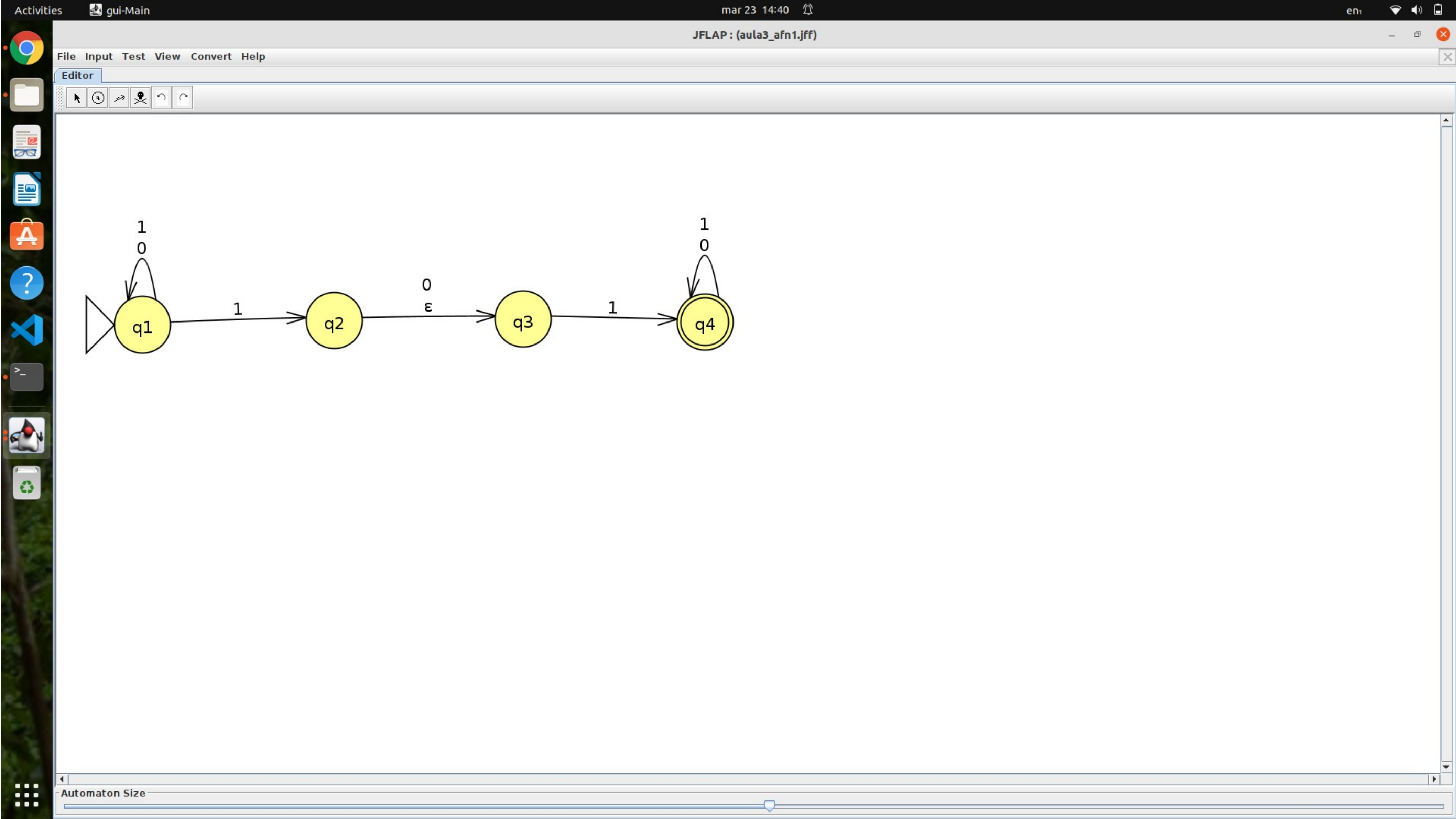
http://www.marcusramos.com.br/univasf/livro-lfa/index.html

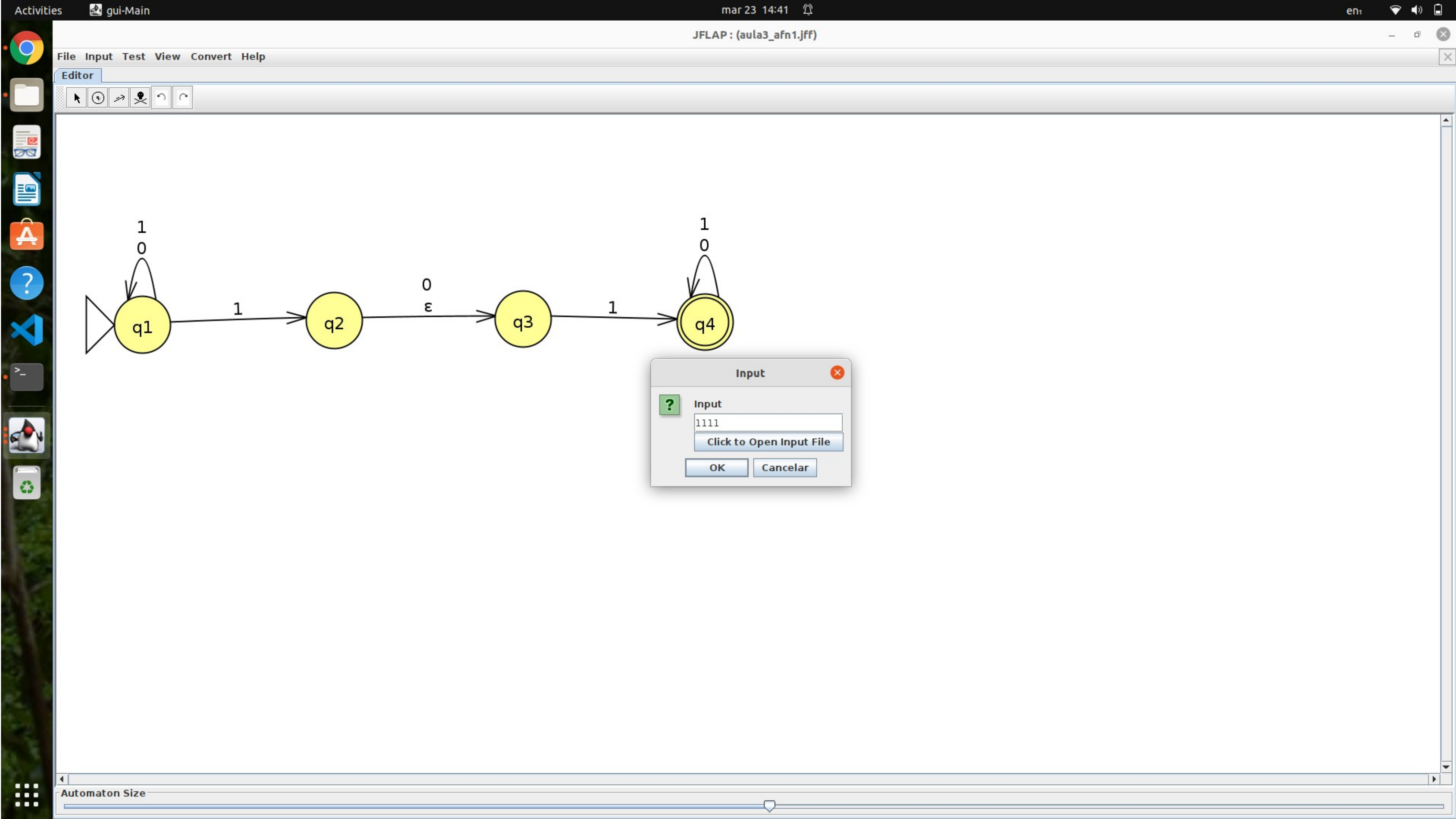


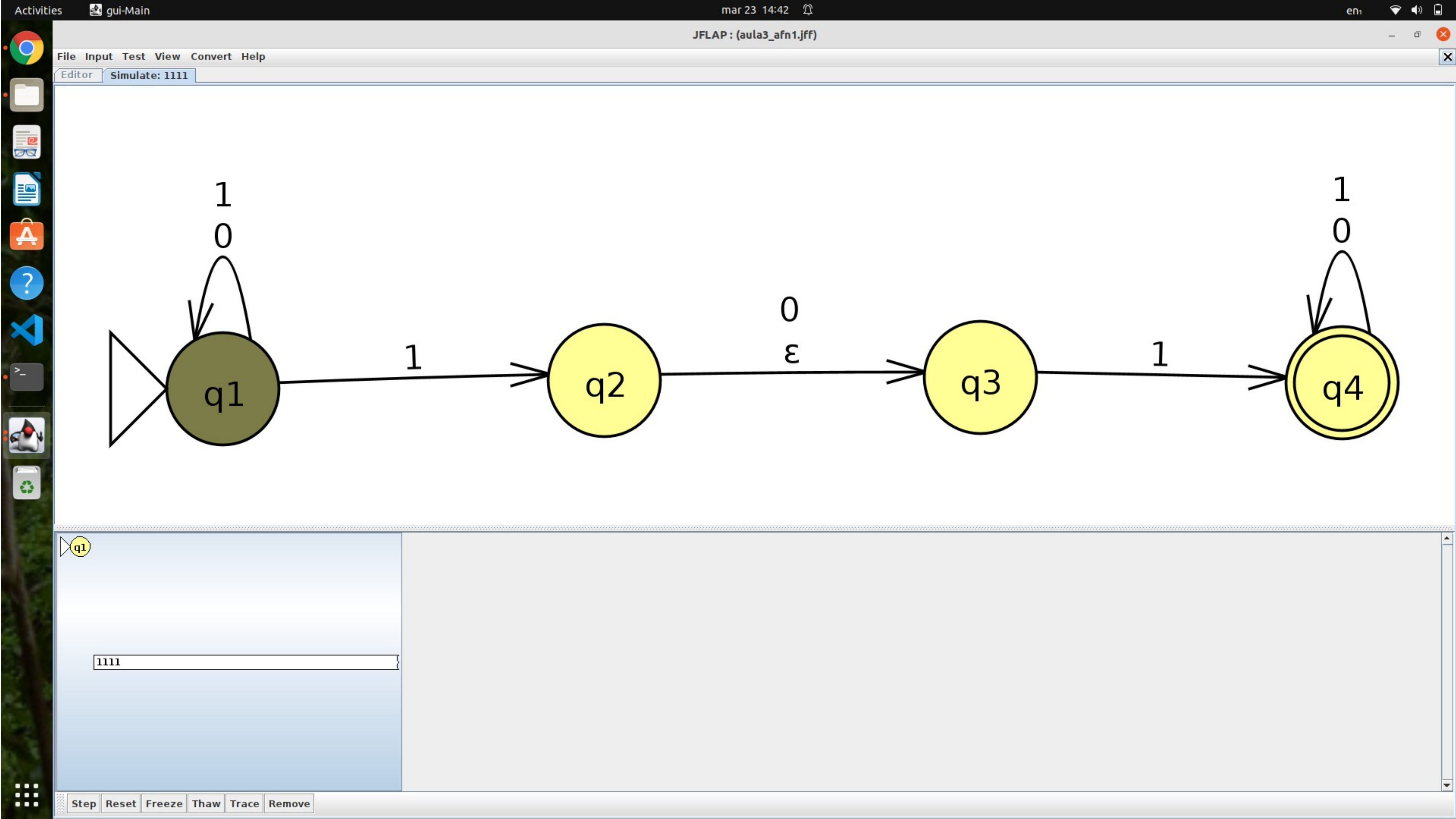
Exercício: façam no papel e no JFlap a simulação para a entrada 1111

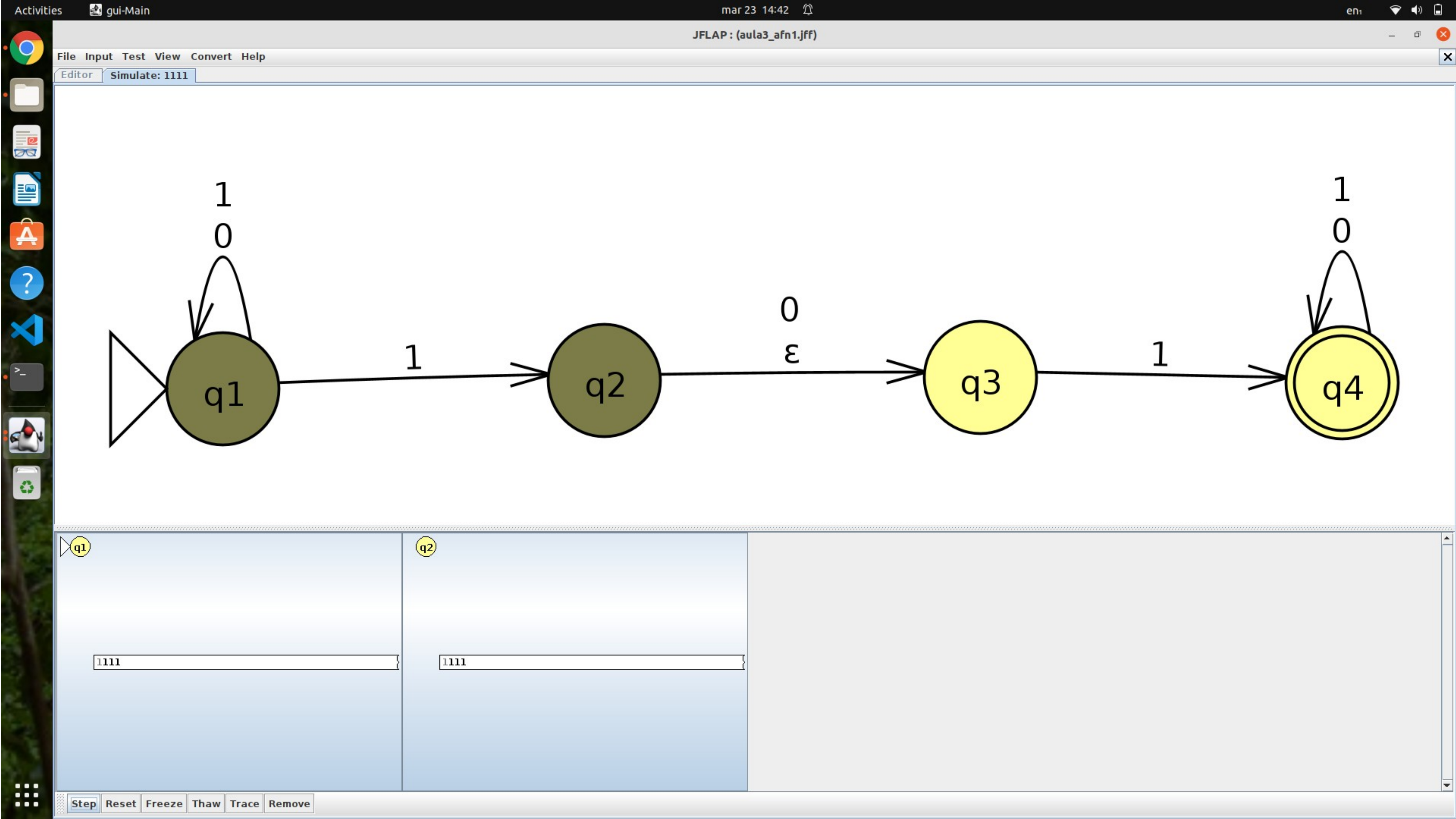


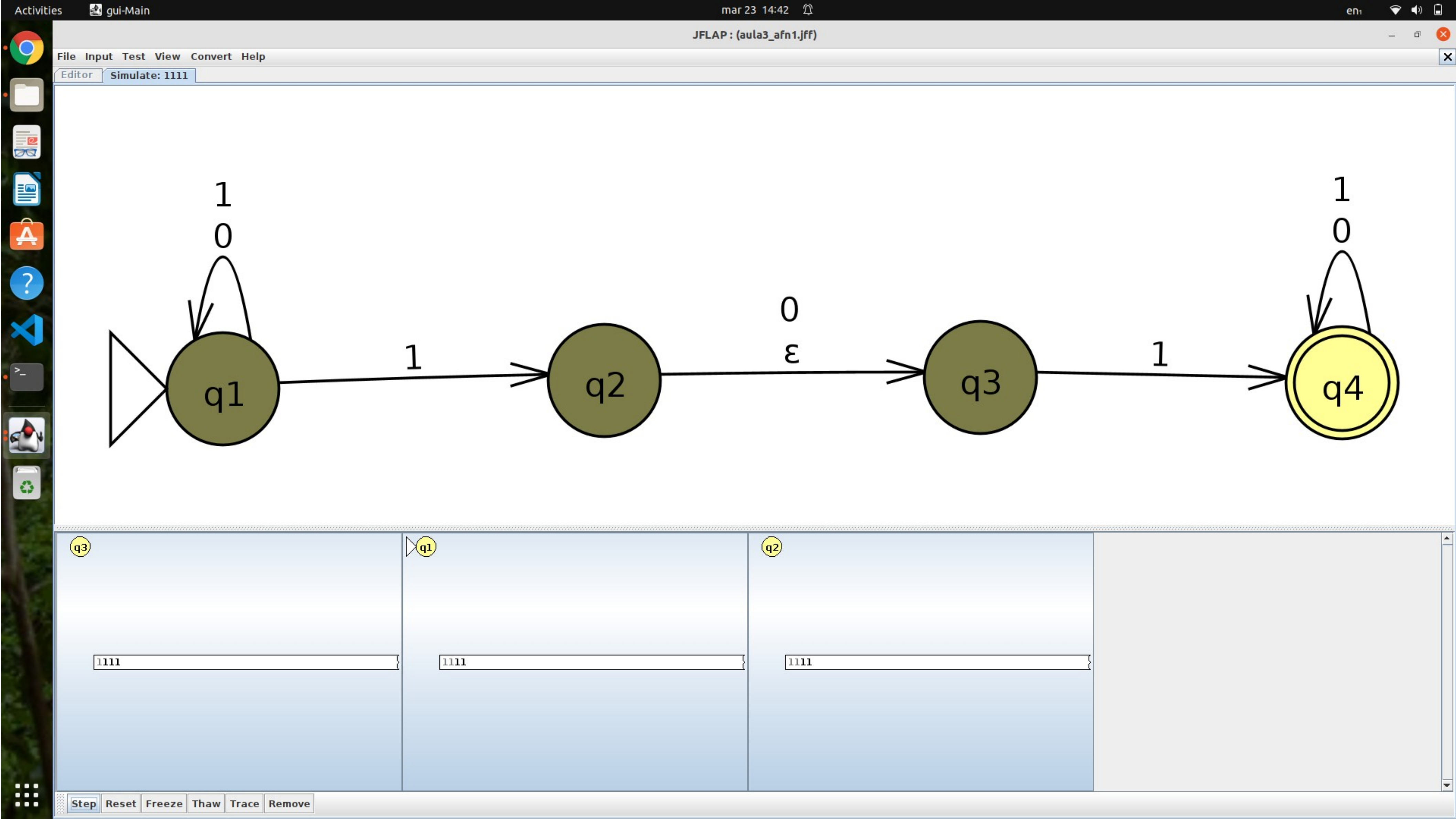

Activities
gui-Main
mar 23 14:40
en
JFLAP 7.1
File
Help
Batch
Preferences
Finite Automaton
Mealy Machine
Moore Machine
Pushdown Automaton
Turing Machine
Multi-Tape Turing Machine
Turing Machine With Building Blocks
Grammar
L-System
Regular Expression
Regular Pumping Lemma
Context-Free Pumping Lemma
Ian Goodfellow, Yoshua Bengio, ...
agendaDiaria.ods
Obsidian-0.13.23.AppImage

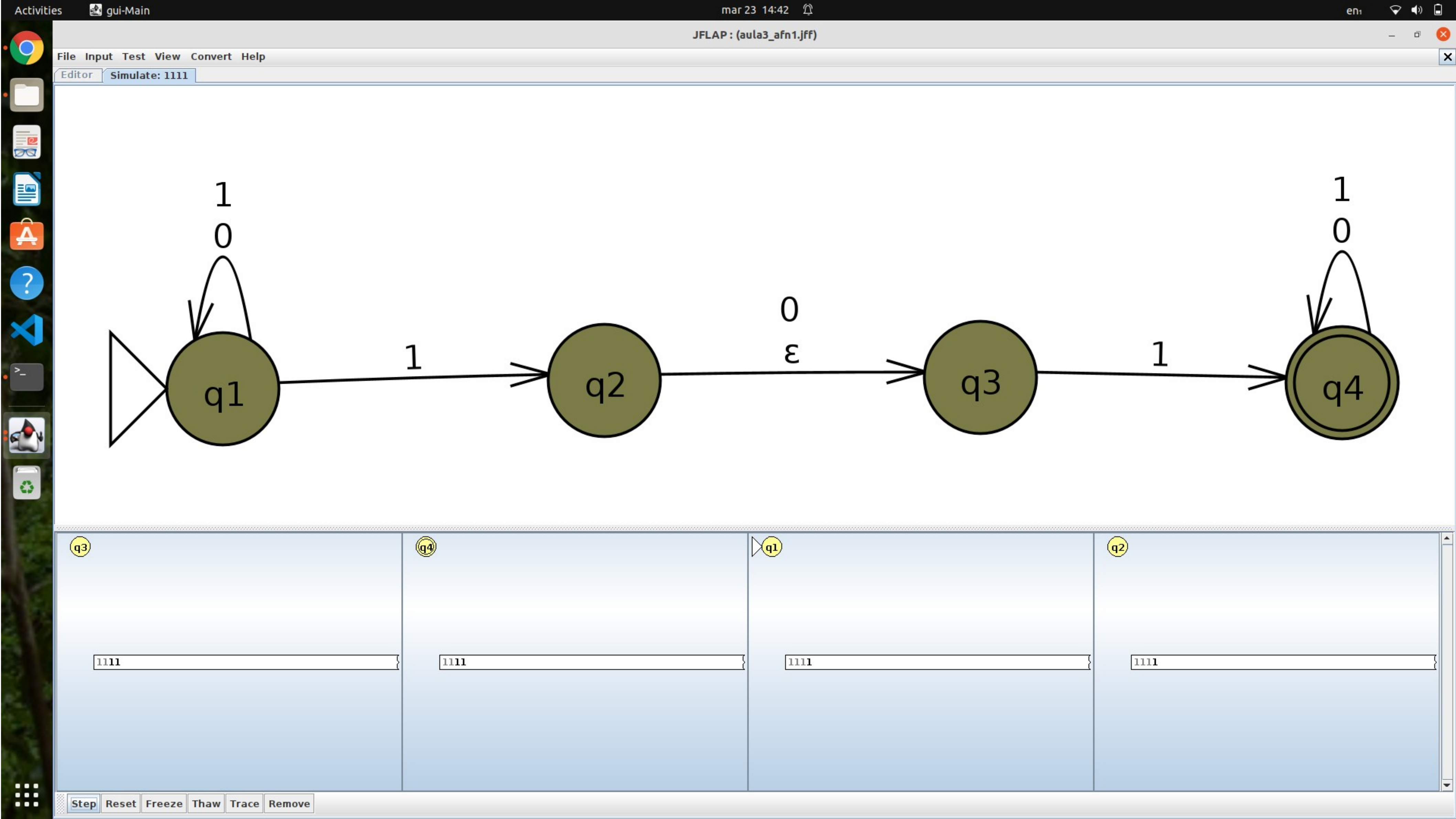

Activities
gui-Main
mar 23 14:40
JFLAP : (aula3_afn1.jff)
File Input Test View Convert Help
Editor
1
0
q1
1
q2
0
ε
q3
1
q4
1
0
Automaton Size

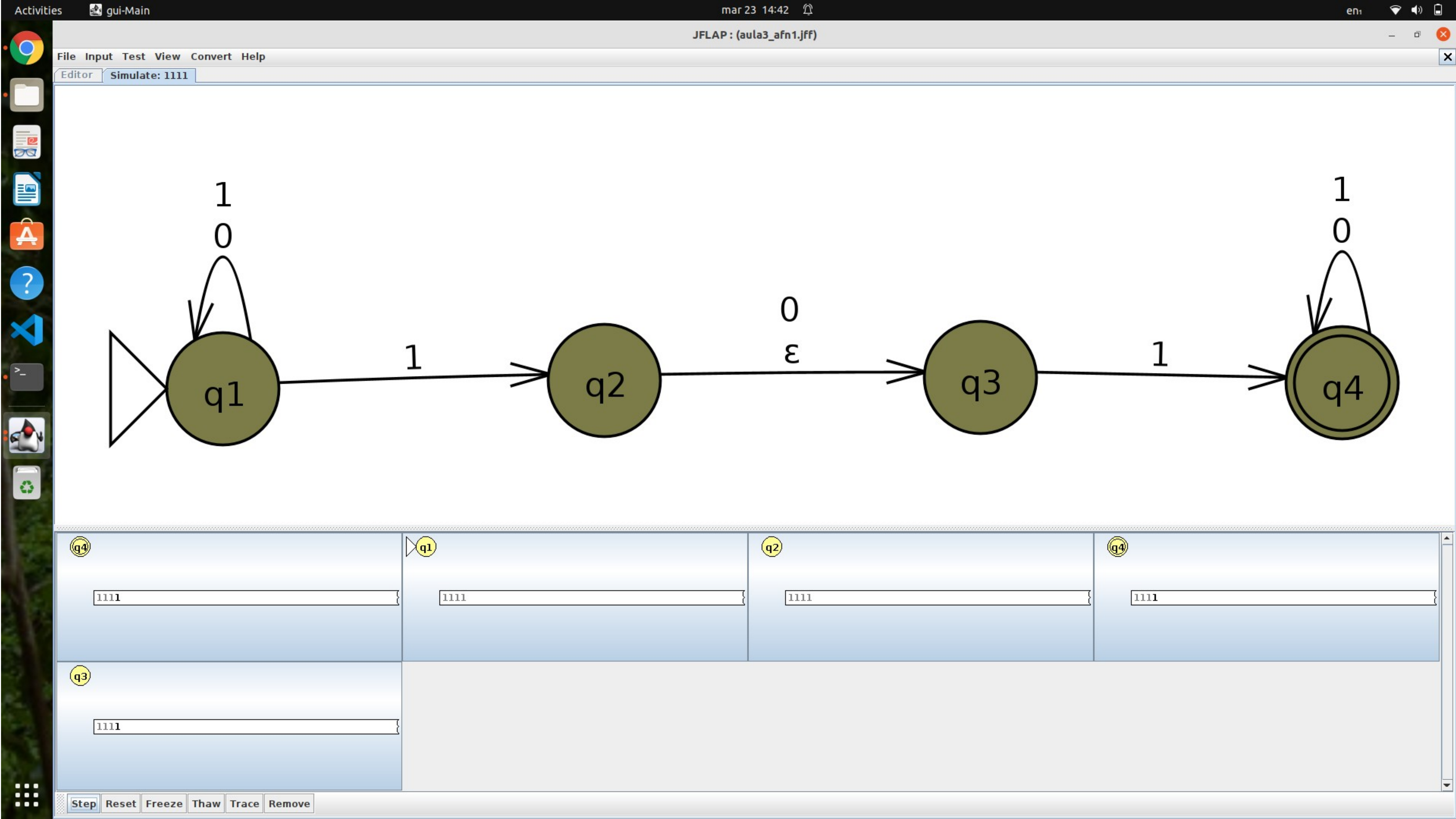

Activities
gui-Main
mar 23 14:41
JFLAP : (aula3_afn1.jff)
File Input Test View Convert Help
Editor
1
0
q1
1
q2
0
ε
q3
1
q4
1
0
Input
Input
1111
Click to Open Input File
OK
Cancelar
Automaton Size

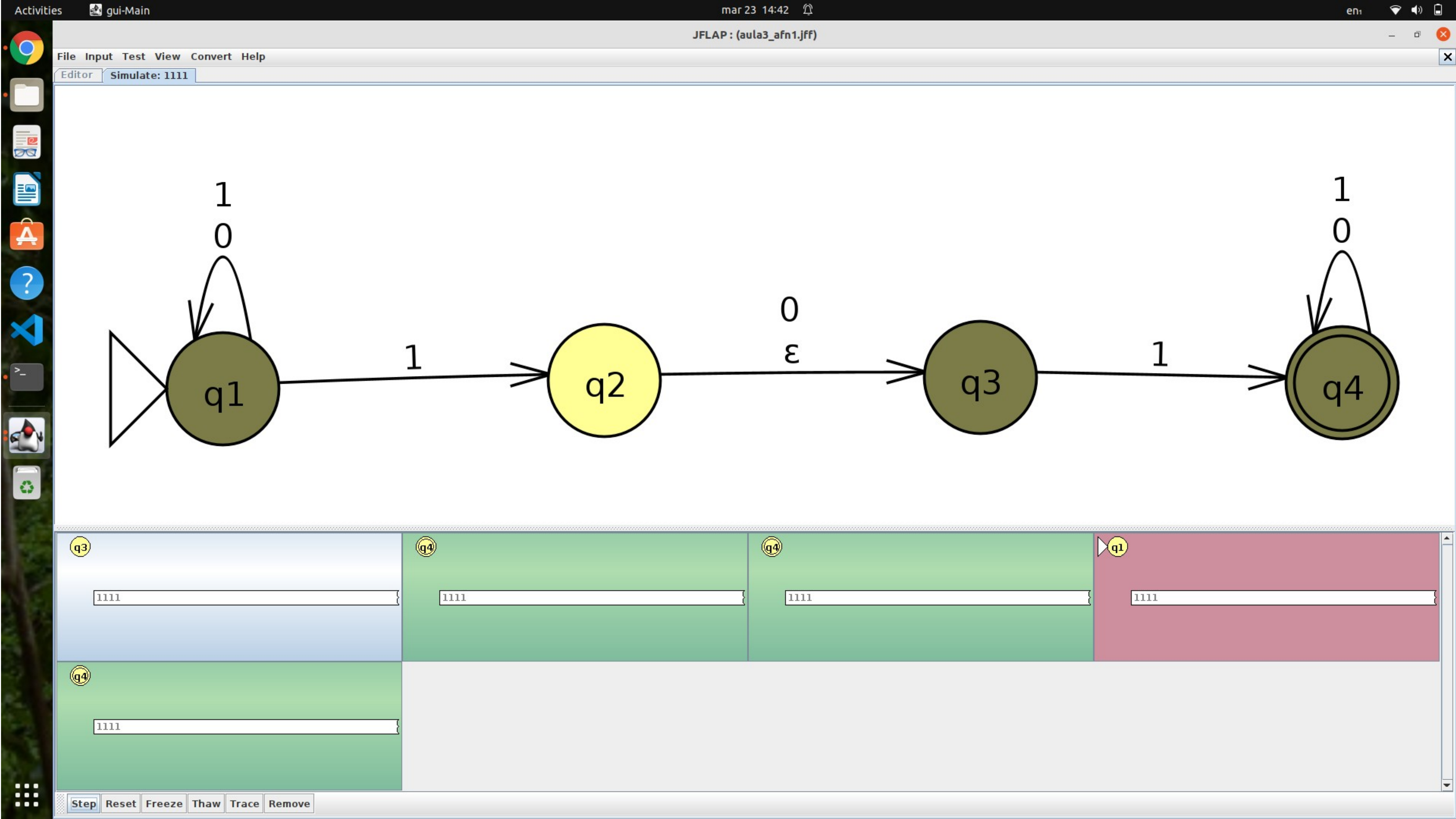

Activities
gui-Main
mar 23 14:42
en
JFLAP : (aula3_afn1.jff)
File
Input
Test
View
Convert
Help
Editor
Simulate: 1111
1
0
1
0
ε
1
1
0
q1
q2
q3
q4
q1
1111
Step
Reset
Freeze
Thaw
Trace
Remove

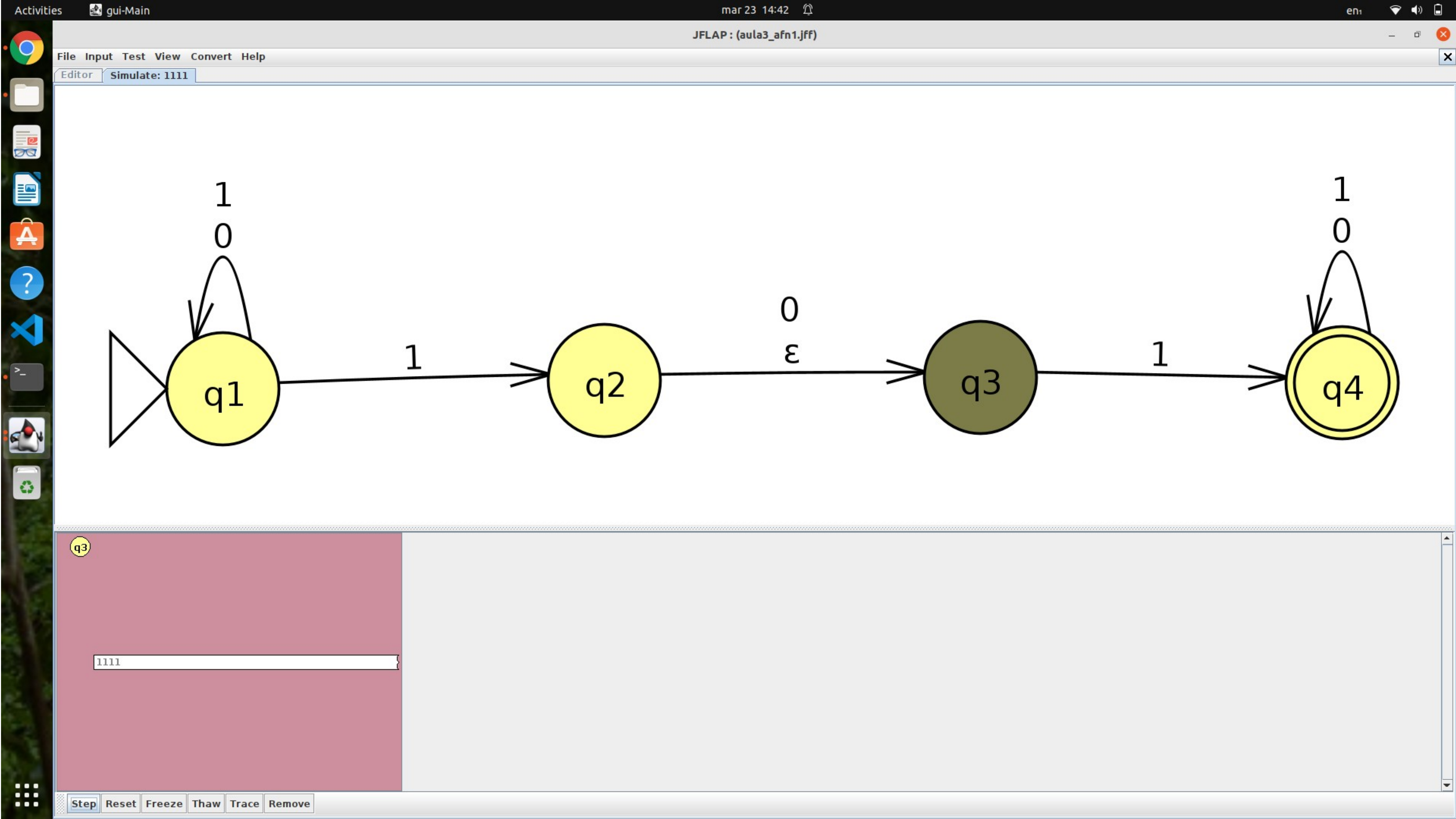

Activities
gui-Main
mar 23 14:42
en
JFLAP : (aula3_afn1.jff)
File Input Test View Convert Help
Editor
Simulate: 1111
1
0
1
0
ε
1
1
0
q1
q2
q3
q4
q1
q2
1111
1111
Step Reset Freeze Thaw Trace Remove


Activities
gui-Main
mar 23 14:42
en
JFLAP : (aula3_afn1.jff)
File
Input
Test
View
Convert
Help
Editor
Simulate: 1111
1
0
1
0
ε
1
1
0
q1
q2
q3
q4
q3
q1
q2
1111
1111
1111
Step
Reset
Freeze
Thaw
Trace
Remove


Activities
gui-Main
mar 23 14:42
JFLAP : (aula3_afn1.jff)
File Input Test View Convert Help
Editor
Simulate: 1111
1
0
1
0
1
0
ε
1
q1
q2
q3
q4
q3
q4
q1
q2
1111
1111
1111
1111
Step Reset Freeze Thaw Trace Remove


Activities
gui-Main
mar 23 14:42
en
JFLAP : (aula3_afn1.jff)
File Input Test View Convert Help
Editor
Simulate: 1111
1
0
1
0
ε
1
1
0
q1
q2
q3
q4
q4
1111
q1
1111
q2
1111
q4
1111
q3
1111
Step Reset Freeze Thaw Trace Remove


Activities
gui-Main
mar 23 14:42
en
JFLAP : (aula3_afn1.jff)
File
Input
Test
View
Convert
Help
Editor
Simulate: 1111
1
0
q1
1
q2
0
ε
q3
1
1
0
q4
q3
1111
q4
1111
q4
1111
q1
1111
q4
1111
Step
Reset
Freeze
Thaw
Trace
Remove


Activities
gui-Main
mar 23 14:42
en
JFLAP : (aula3_afn1.jff)
File
Input
Test
View
Convert
Help
Editor
Simulate: 1111
1
0
q1
1
q2
0
ε
q3
1
q4
1
0
q3
1111
Step
Reset
Freeze
Thaw
Trace
Remove

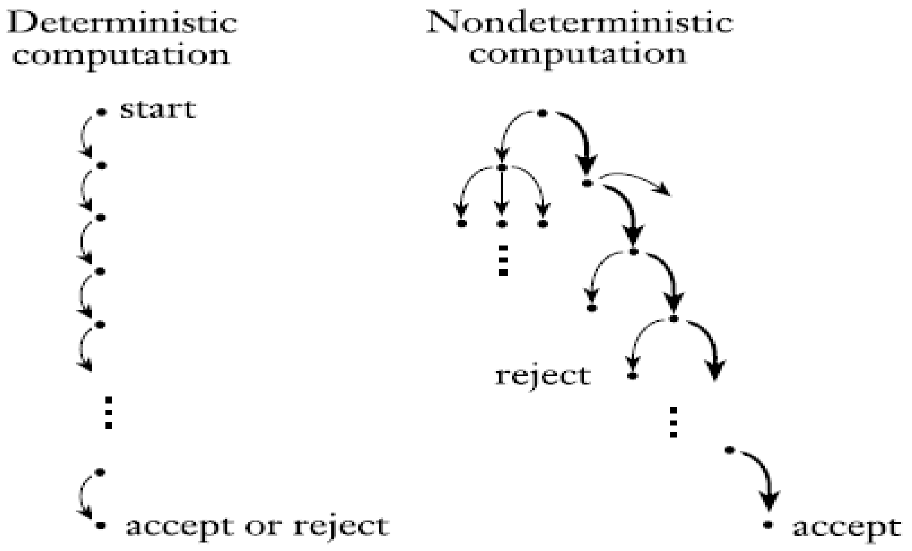
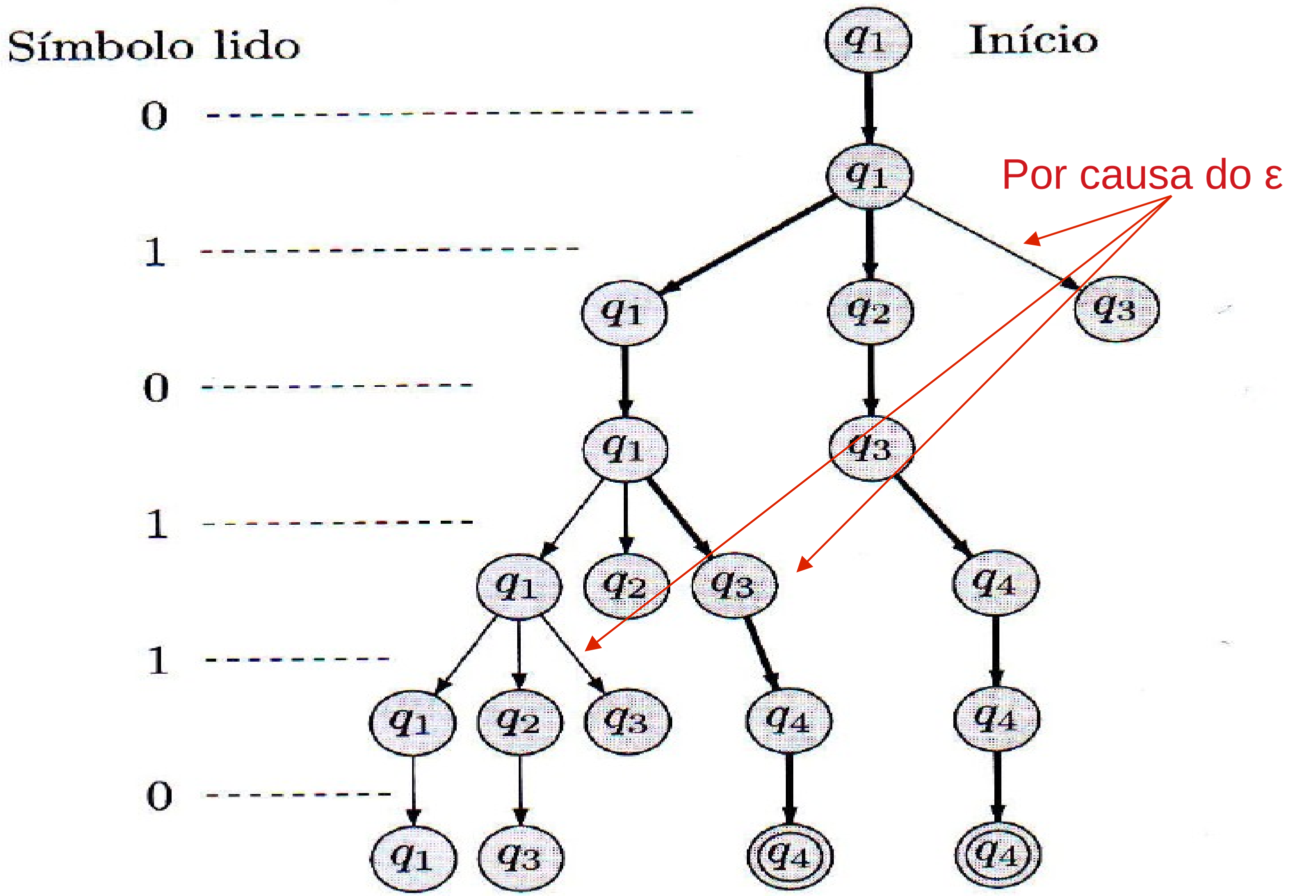


FIGURA 1.28
Computações determinísticas e não-determinísticas com um ramo de aceitação

Logo, quem são mais eficientes? (em termos de tempo...)

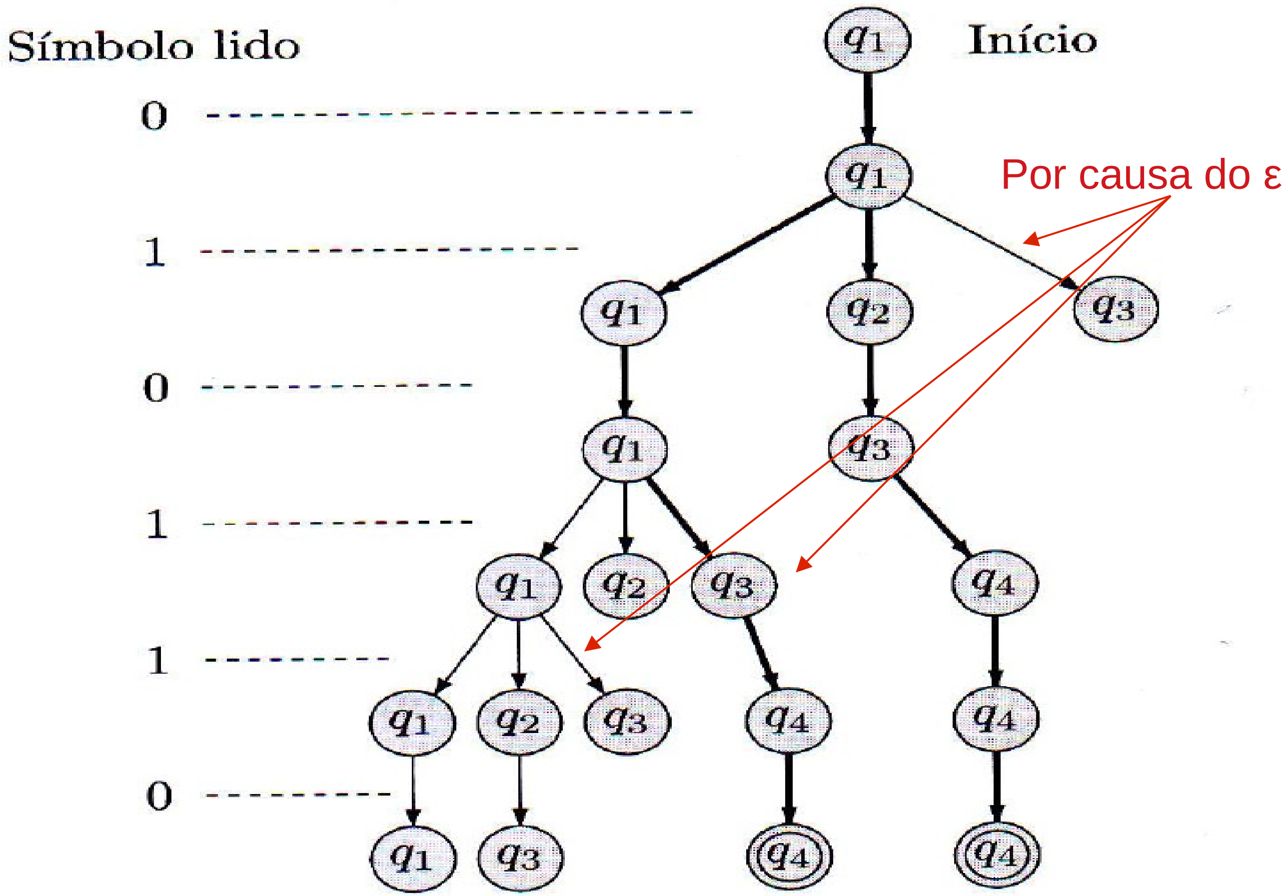
Qual a complexidade (tempo) de análise de uma cadeia por um **AFN**?



Qual a complexidade (tempo) de análise de uma cadeia por um **AFN**?

O(n * |Q|)

Cada “nó” dessa árvore poderia ter no máximo |Q| filhos, e poderíamos descartar nós repetidos em um mesmo nível



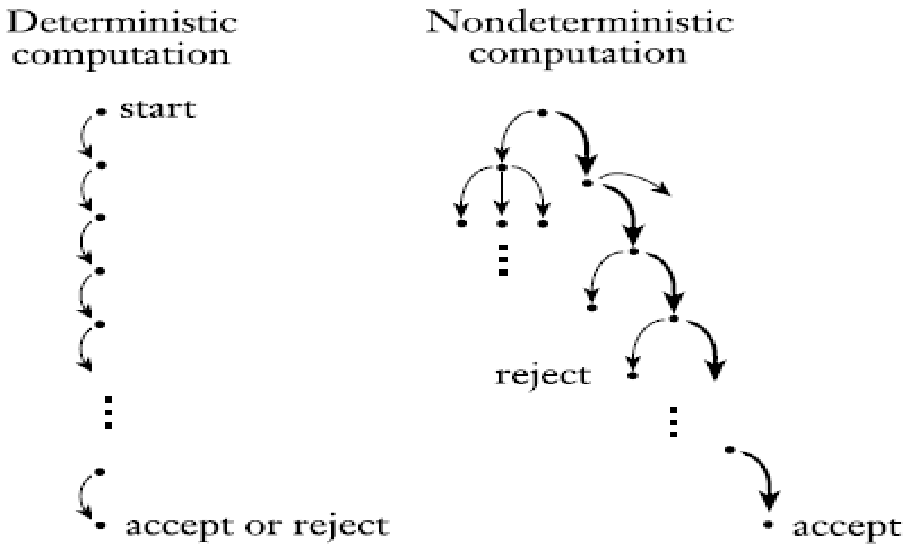


FIGURA 1.28
Computações determinísticas e não-determinísticas com um ramo de aceitação

Logo, quem são mais eficientes? (em termos de tempo...)

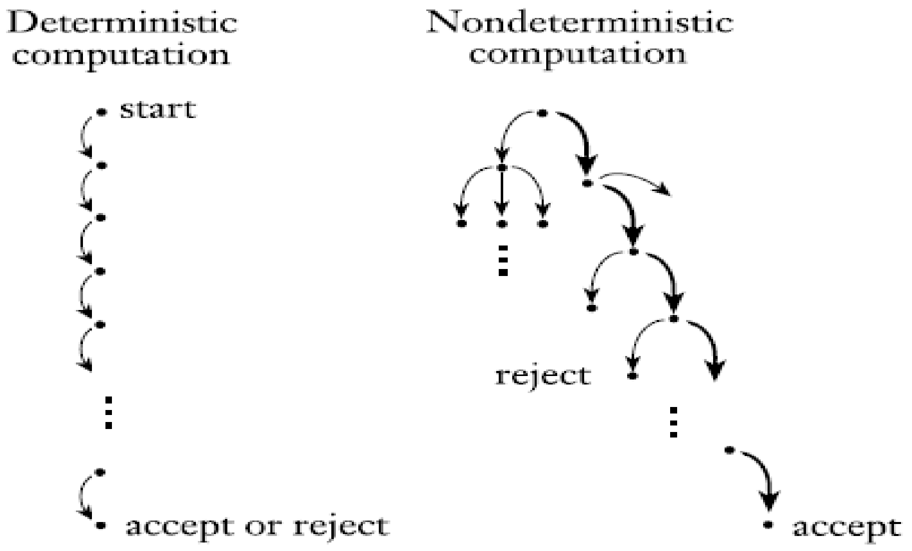


FIGURA 1.28
Computações determinísticas e não-determinísticas com um ramo de aceitação

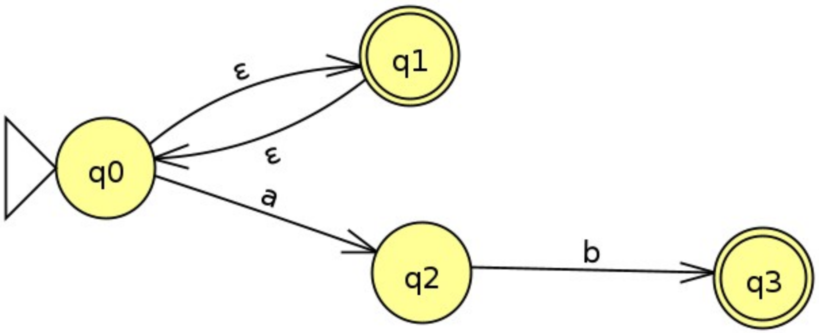
Logo, quem são mais eficientes? (em termos de tempo...)

AFD's são mais eficientes que AFN's (tempo)

Perigo das transições no vazio

- M = …
  - $\delta = \{(q_0, \varepsilon) \rightarrow q_1, (q_0, a) \rightarrow q_2, (q_1, \varepsilon) \rightarrow q_0, (q_2, b) \rightarrow q_3\}$
  - $F = \{q_1, q_3\}$

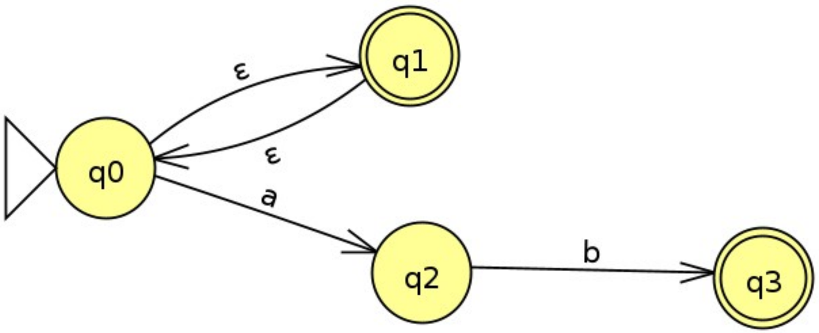
entrada: aa



Perigo das transições no vazio

- M = ...
  - $\delta = \{(q_0, \varepsilon) \to q_1, (q_0, a) \to q_2, (q_1, \varepsilon) \to q_0, (q_2, b) \to q_3\}$
  - $F = \{q_1, q_3\}$

  entrada: aa



- O autômato não pára!

- Felizmente há um algoritmo para eliminação de transições no vazio

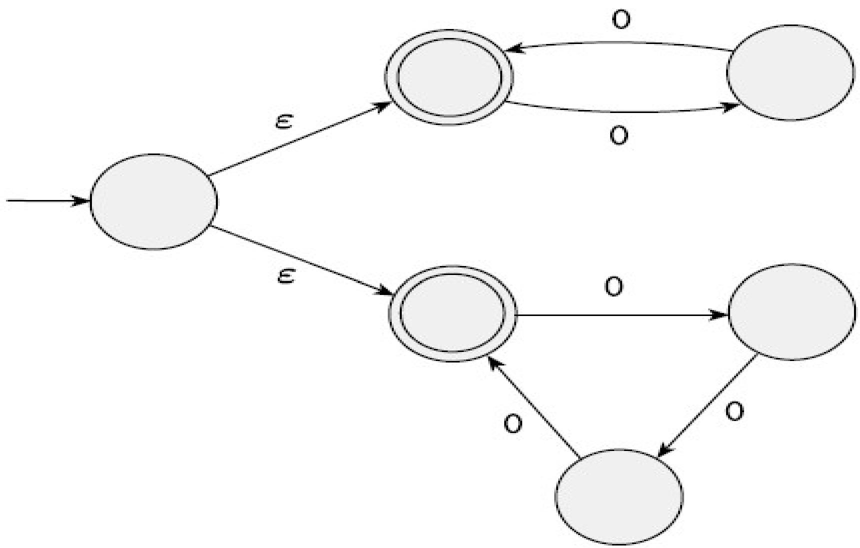
Exercício

- Desenhe um AFN para a linguagem formada por sequências formadas apenas por zeros, mas que contenham o nr de zeros sendo um múltiplo de 2 ou múltiplo de 3

Exercício

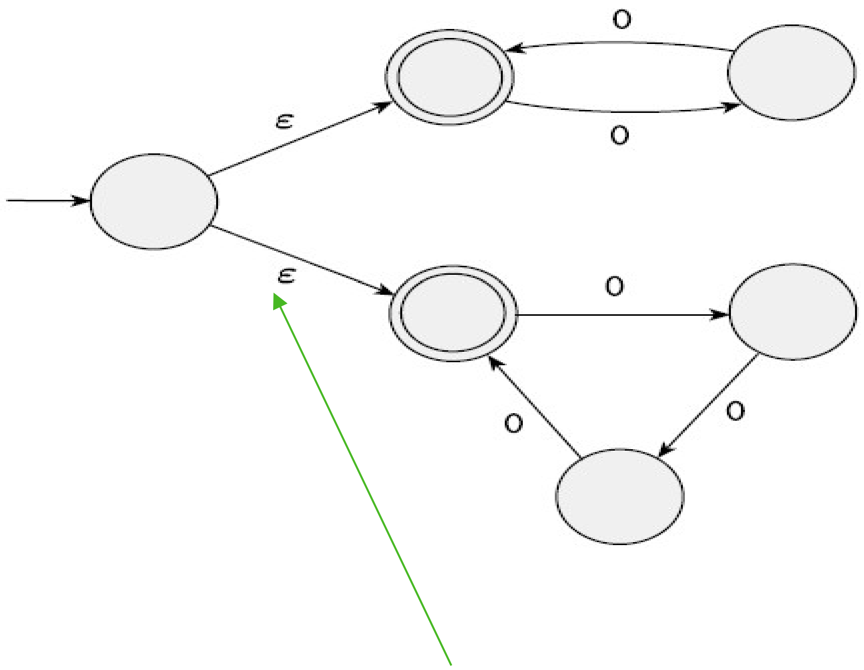
- Desenhe um AFN para a linguagem formada por sequências formadas apenas por zeros, mas que contenham o nr de zeros sendo um múltiplo de 2 **ou** múltiplo de 3

Exercício - Resposta



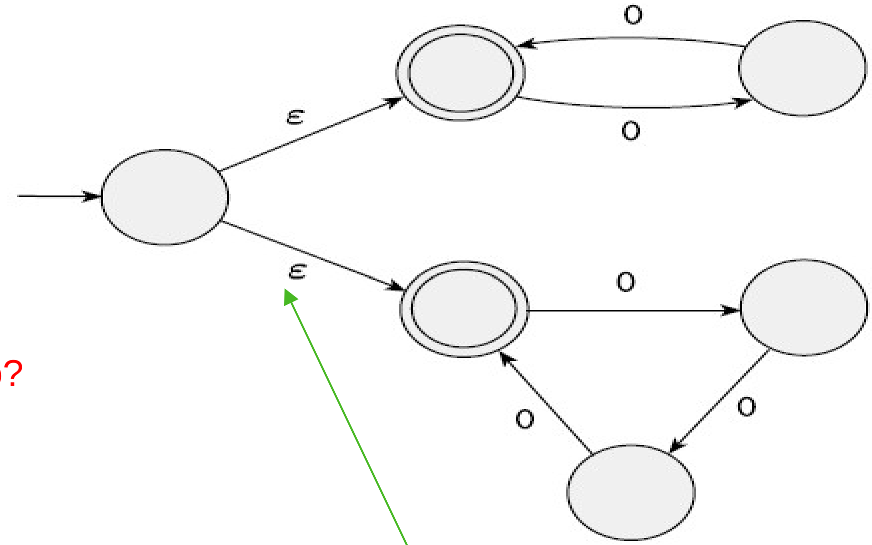
L = ?

Exercício - Resposta



L = { w | w ϵ 0* e |w| = 2*i, i = 0, 1, …} U { w | w ϵ 0* e |w| = 3*i, i = 0, 1, …}

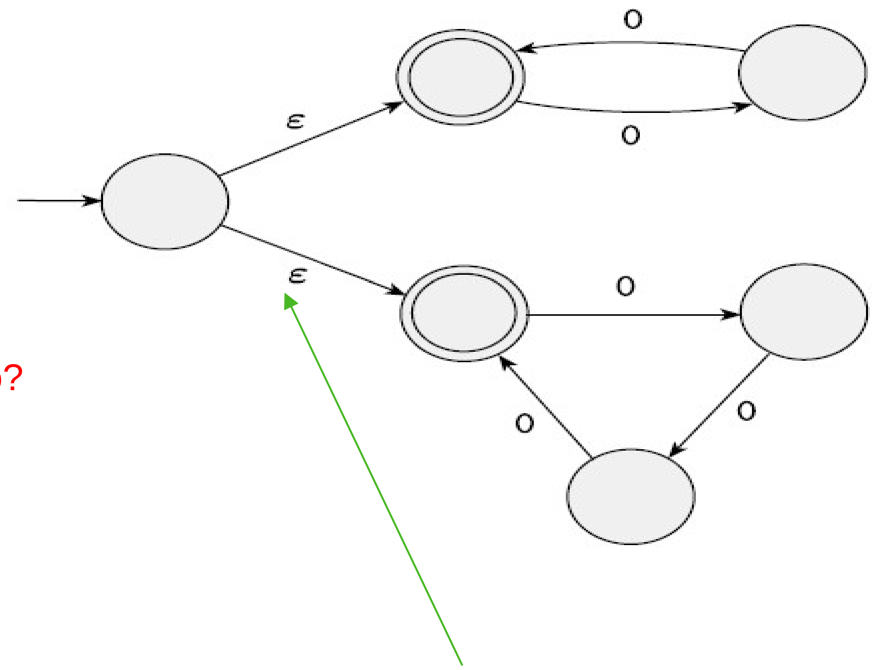
Exercício - Resposta



Onde está o não determinismo?

$L = \{ w \mid w \in 0^* \text{ e } |w| = 2*i, i = 0, 1, \ldots\} \cup \{ w \mid w \in 0^* \text{ e } |w| = 3*i, i = 0, 1, \ldots\}$

Exercício - Resposta



Onde está o não determinismo?

Nas duas arestas com ε

L = { w | w ϵ 0* e |w| = 2*i, i = 0, 1, …} U { w | w ϵ 0* e |w| = 3*i, i = 0, 1, …}

Exercícios

1) Desenhe um AFN para a linguagem formada por sequências binárias que contenham 1 na antepenúltima posição

2) Desenhe o diagrama de estados de um AFN que reconheça a linguagem $1^*01^+0^*$

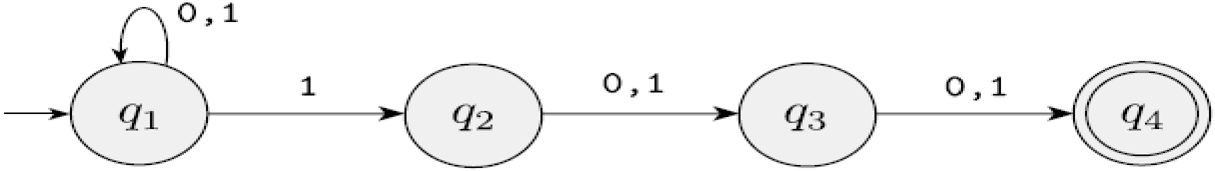
Exercícios

1) Desenhe um AFN para a linguagem formada por sequências binárias que contenham 1 na antepenúltima posição

Faça antes de olhar o próximo slide!!!

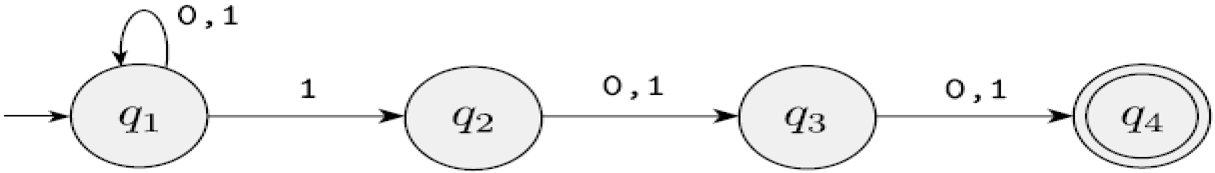
Exercícios

1) Desenhe um AFN para a linguagem formada por sequências binárias que contenham 1 na antepenúltima posição



Exercícios

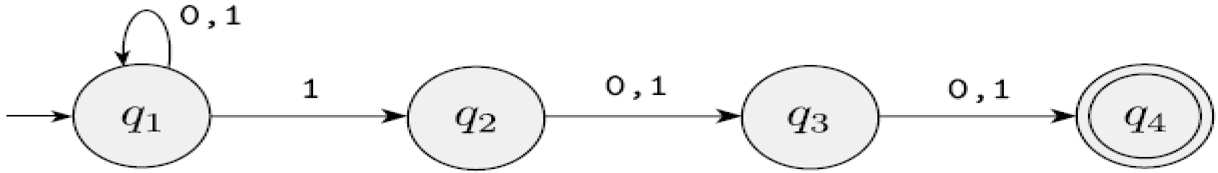
1) Desenhe um AFN para a linguagem formada por sequências binárias que contenham 1 na antepenúltima posição



Onde está o não determinismo?

Exercícios

1) Desenhe um AFN para a linguagem formada por sequências binárias que contenham 1 na antepenúltima posição



Onde está o não determinismo?

q1 tem duas opções de próximo estado se ler o símbolo “1”

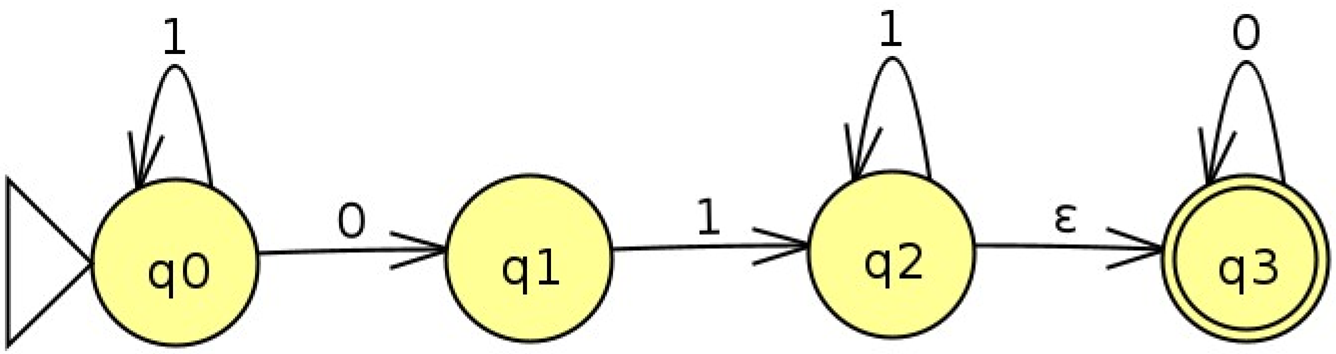
Exercícios

2) Desenhe o diagrama de estados de um AFN que reconheça a linguagem $1^*01^+0^*$

Faça antes de olhar o próximo slide!!!

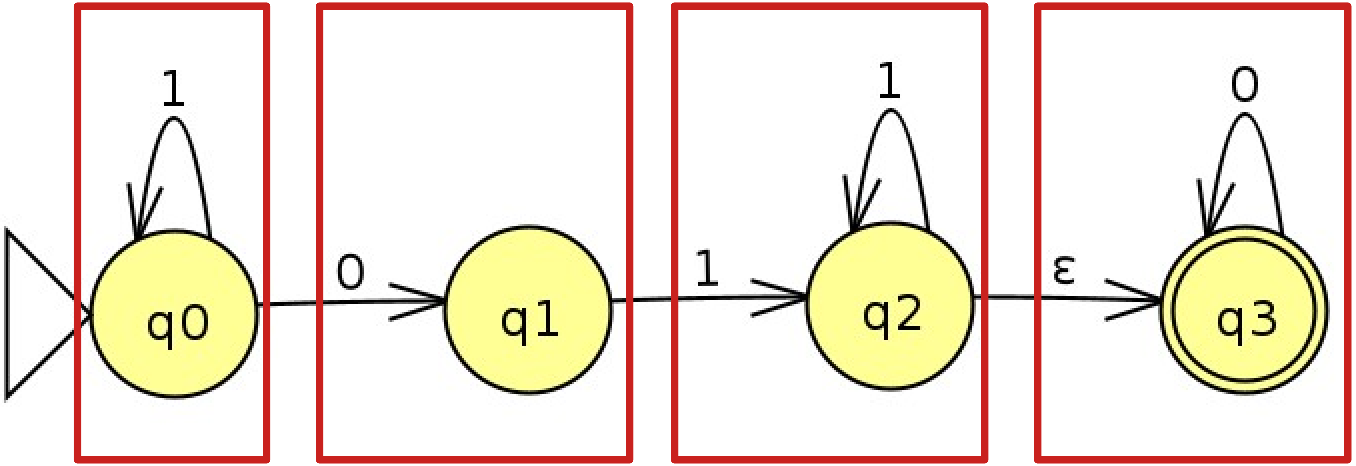
Exercícios

2) Desenhe o diagrama de estados de um AFN que reconheça a linguagem $1^*01^+0^*$



Exercícios

2) Desenhe o diagrama de estados de um AFN que reconheça a linguagem $1^*01^+0^*$



1^* (no estado inicial)

0

1^+

0^*

Lista de Exercícios do Sipser (2ª ed)

Exercícios 1.1, 1.2, 1.3, 1.5, 1.6 e 1.7

No 1.7:

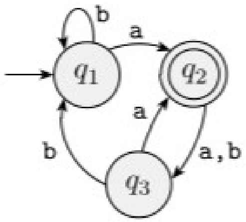
* : 0 ou mais vezes

+ : 1 ou mais vezes

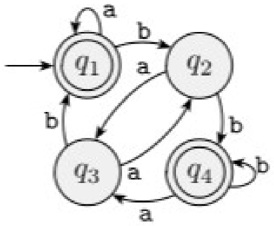
Ex: $1^*(001^+)^*$

Lista mínima da mínima (o que está em vermelho). Aconselhável fazer todos.

[R]**1.1** The following are the state diagrams of two AFDs, M_1 and M_2. Answer the following questions about each of these machines.



M_1



M_2

a. What is the start state?

b. What is the set of accept states?

c. What sequence of states does the machine go through on input aabb?

d. Does the machine accept the string aabb?

e. Does the machine accept the string ε?

[R]**1.2** Give the formal description of the machines M_1 and M_2 pictured in Exercise 1.1.

1.3 The formal description of a AFD M is $(\{q_1, q_2, q_3, q_4, q_5\}, \{u, d\}, \delta, q_3, \{q_3\})$, where δ is given by the following table. Give the state diagram of this machine.

	u	d
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_2	q_4
q_4	q_3	q_5
q_5	q_4	q_5

1.5 Each of the following languages is the complement of a simpler language. In each part, construct a AFD for the simpler language, then use it to give the state diagram of a AFD for the language given. In all parts $\Sigma = \{a, b\}$.

[R]**a.** $\{w \mid w$ does not contain the substring ab$\}$

[R]**b.** $\{w \mid w$ does not contain the substring baba$\}$

Difícil por enquanto… espere a aula sobre "fechamentos" se não conseguir

c. $\{w \mid w$ contains neither the substrings ab nor ba$\}$

d. $\{w \mid w$ is any string not in $a^*b^*\}$

e. $\{w \mid w$ is any string not in $(ab^+)^*\}$

f. $\{w \mid w$ is any string not in $a^* \cup b^*\}$

g. $\{w \mid w$ is any string that doesn't contain exactly two a's$\}$

h. $\{w \mid w$ is any string except a and b$\}$

1.6 Give state diagrams of AFDs recognizing the following languages. In all parts the alphabet is $\{0,1\}$.

a. $\{w \mid w$ begins with a 1 and ends with a 0$\}$

b. $\{w \mid w$ contains at least three 1s$\}$

c. $\{w \mid w$ contains the substring 0101, i.e., $w = x0101y$ for some x and $y\}$

d. $\{w \mid w$ has length at least 3 and its third symbol is a 0$\}$

e. $\{w \mid w$ starts with 0 and has odd length, or starts with 1 and has even length$\}$

f. $\{w \mid w$ doesn't contain the substring 110$\}$

g. $\{w \mid$ the length of w is at most 5$\}$

h. $\{w \mid w$ is any string except 11 and 111$\}$

i. $\{w \mid$ every odd position of w is a 1$\}$

j. $\{w \mid w$ contains at least two 0s and at most one 1$\}$

k. $\{\varepsilon, 0\}$

l. $\{w \mid w$ contains an even number of 0s, or contains exactly two 1s$\}$

m. The empty set

n. All strings except the empty string

1.7 Give state diagrams of AFNs with the specified number of states recognizing each of the following languages. In all parts the alphabet is $\{0,1\}$.

[R]**a.** The language $\{w \mid w$ ends with 00$\}$ with three states

b. The language of Exercise 1.6c with five states

c. The language of Exercise 1.6l with six states

d. The language $\{0\}$ with two states

e. The language $0^*1^*0^+$ with three states

[R]**f.** The language $1^*(001^+)^*$ with three states

g. The language $\{\varepsilon\}$ with one state

h. The language 0^* with one state