FUNDAMENTOS TEÓRICOS DA COMPUTAÇÃO

--- LINGUAGEM LIVRE DE CONTEXTO ---

Autômatos com Pilhas

Introdução

Autômatos com Pilha (AP) são basicamente uma **AF** com uma pilha (em cada cópia). Pode ser determinístico (APD) ou **não-determinístico** (APND). **Os APND são mais comuns**

É **outro modelo** de máquina de estados que permite representar um determinado conjunto de linguagens, i.e., as **linguagens livres de contexto**

Os APs **possuem uma memória** capaz de registrar alguma informação sobre os passos anteriores da execução da máquina

Esta memória é uma **pilha de símbolos**, de onde elementos podem ser consultados, retirados (desempilhados) e empilhados a cada transição

Introdução

A pilha é uma **memória extra "ilimitada"** que funciona no esquema "*último a entrar, primeiro a sair*" (LIFO – ***Last in First Out***)

Pilhas nos dão o poder de **recursão**

Os símbolos que podem ser empilhados e desempilhados fazem parte de um **alfabeto próprio**, chamado de alfabeto da pilha e é designado pela letra Γ (gama)

O alfabeto da pilha Γ é distinto do alfabeto de entrada Σ

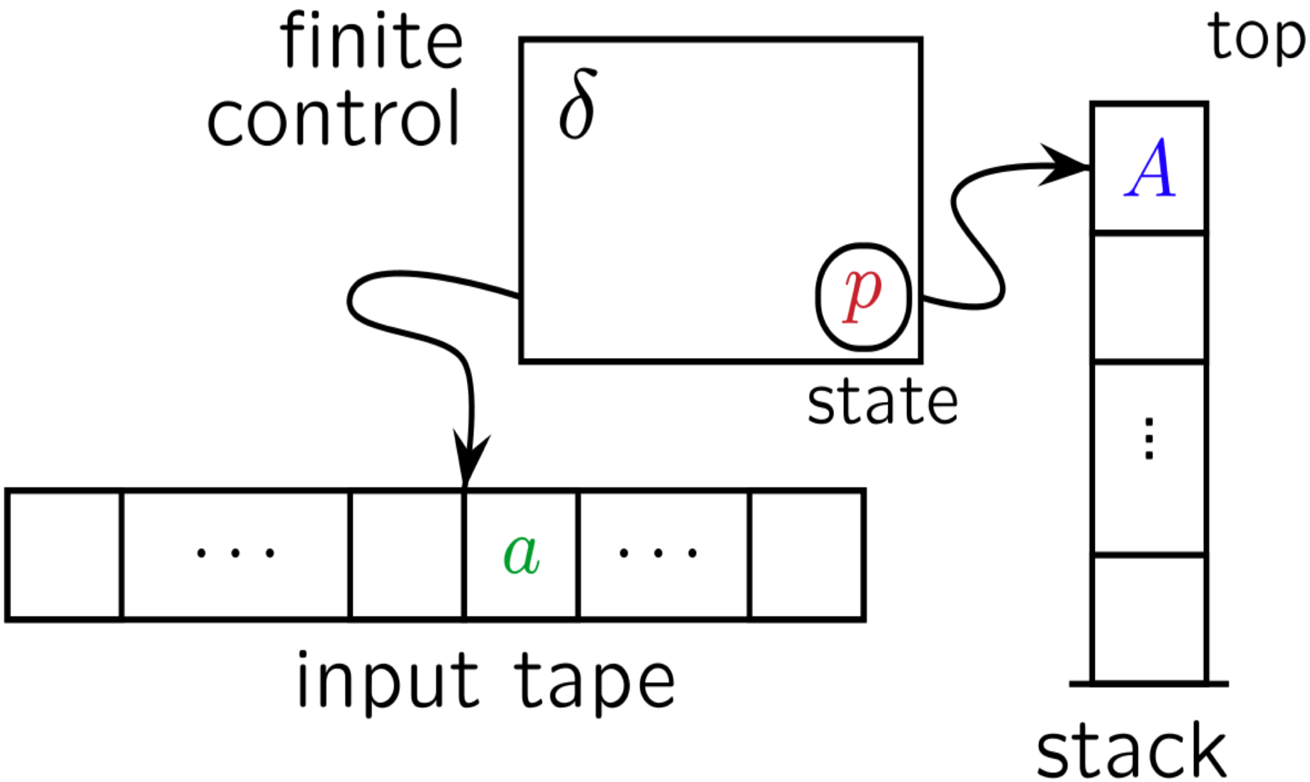
Definição Formal

Um autômato de pilha (AP) é uma 6-tupla $(Q, \Sigma, \Gamma, \delta, q_0, F)$, onde:

- Q é um conjunto finito de estados
- Σ é o alfabeto de entrada
- Γ é o alfabeto da pilha
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$ é a função de transição

  $\boldsymbol{\delta(q_i, a, X) \rightarrow (q_j, Y)}$
- $q_0 \in Q$ é o estado inicial
- $F \subseteq Q$ é o conjunto de estados de aceitação

Definição Formal

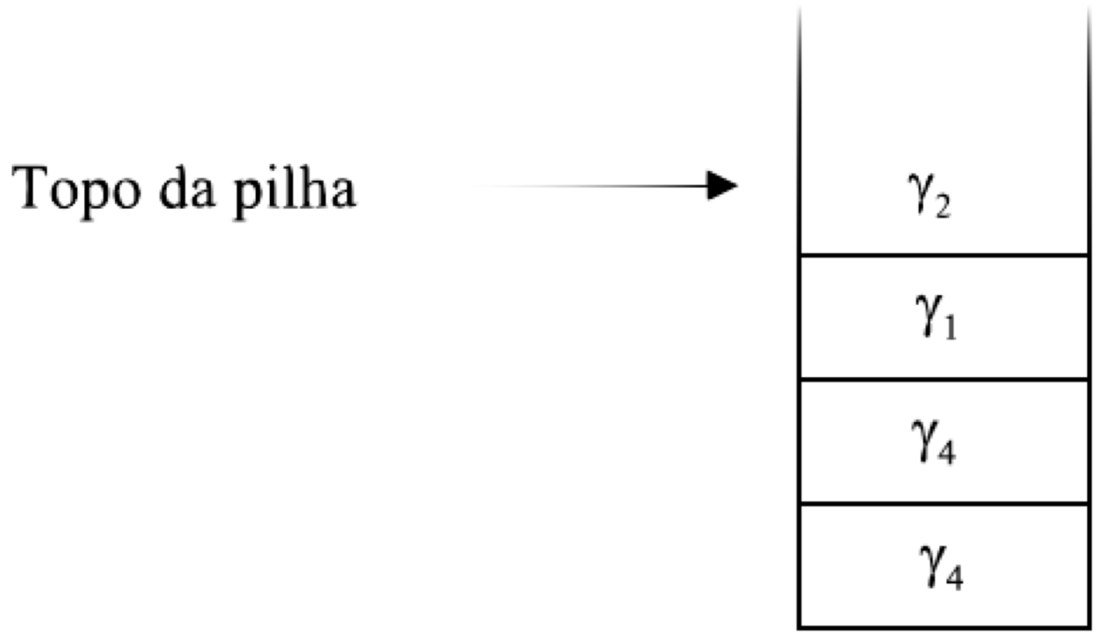


Empilhamento

Considere $\Gamma = \{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$

A cadeia $\gamma_2\gamma_1\gamma_4\gamma_4$, por exemplo, representa o conteúdo da pilha

Note-se que γ_2 se encontra no **topo** da pilha e que γ_4 se encontra no **fundo** da pilha



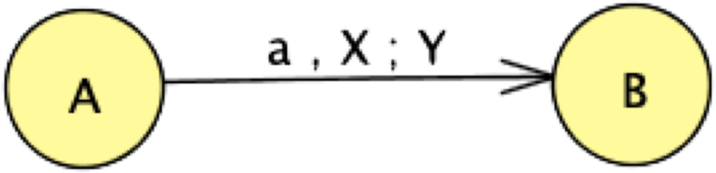
Iniciação e Reconhecimento

- O AP sempre inicia a sua execução com a **pilha vazia**

- O modelo padrão de AP reconhece uma palavra quando as três condições seguintes são obedecidas ***simultaneamente***
  1. a palavra de entrada é inteiramente consumida
  2. a máquina para em um estado de **aceitação** (final)
  3. a **pilha encontra-se vazia** no instante em que a palavra de entrada termina de ser consumida

Transições

Nos APs, as transições representam: (i) o símbolo da palavra consumida (assim como nos AFs); (ii) um símbolo a ser desempilhado (ou ε, caso nada deva ser empilhado); e (iii) uma palavra a ser empilhada



$$\boldsymbol{\delta(A, a, X) \rightarrow (B, Y)}$$

Se o autômato está no estado A e puder consumir o símbolo a, e o símbolo X está no topo da pilha, então ele transitará para o estado B, desempilhará o símbolo X, e empilhará a palavra Y

Para a transição ocorrer é necessário que o símbolo X esteja no topo da pilha

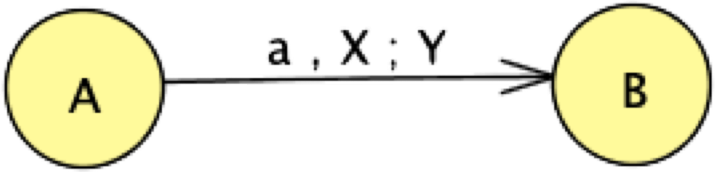
Como reconhecer pilha vazia?

Não é estritamente necessário, mas para não termos muitos problemas com esse reconhecimento, é comum adicionar um símbolo especial como indicador que chegou-se ao fim dos símbolos

Usualmente esse símbolo é o “$”

A ideia é sem ler nada, sem desempilhar nada e empilha o símbolo “$”; no final precisa fazer o processo inverso, isto é, sem ler nada, desempilha o “$” e empilha nada

Transições
:: Exemplo 1



$$\delta(A, a, X) = (B, Y)$$

Supondo:

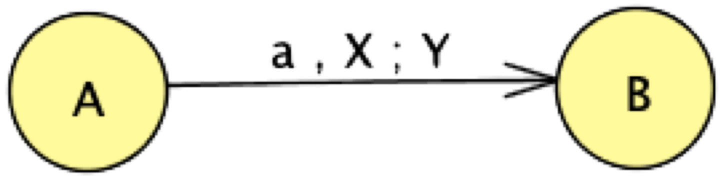
Palavra de entrada $\omega = abcba$;

Estado atual $= A$;

Pilha atual $= XYYX$

A transição ocorre ou não?

Transições
:: Exemplo 1



$\delta(A, a, X) = (B, Y)$

Supondo:

Palavra de entrada $\omega = abcba$;

Estado atual $= A$;

Pilha atual $= XYYX$

Neste caso a transição ocorre

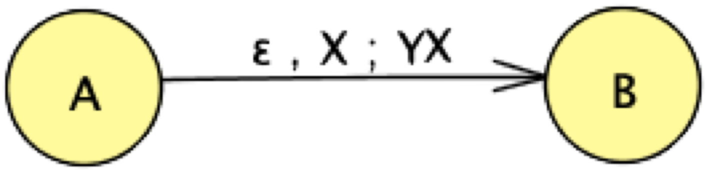
Após a transição...

Palavra de entrada $\omega = bcba$ (consumiu o a);

Estado atual $= B$;

Pilha atual $= \mathrm{Y}YYX$ (desempilhou X e empilhou o Y);

Transições
:: Exemplo 2



$\delta(A, \varepsilon, X) = (B, YX)$

Supondo:

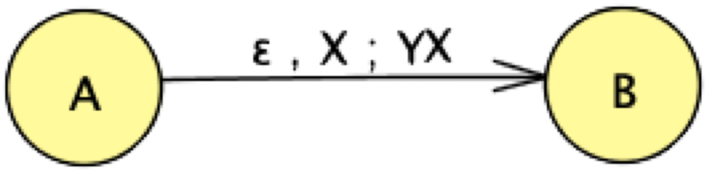
Palavra de entrada $\omega = abcba$;

Estado atual $= A$;

Pilha atual $= XYYX$

A transição ocorre ou não?

Transições :: Exemplo 2



$\delta(A, \varepsilon, X) = (B, YX)$

Supondo:

Palavra de entrada $\omega = abcba$;

Estado atual $= A$;

Pilha atual $= XYYX$

Neste caso a transição ocorre

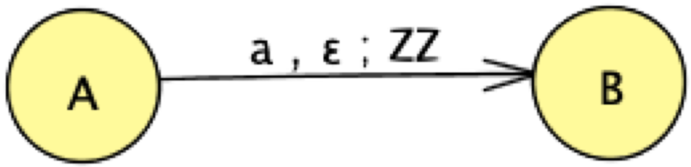
Após a transição...

Palavra de entrada $\omega = abcba$ (consumiu o ε);

Estado atual $= B$;

Pilha atual $=$ $\color{red}{\mathrm{YX}}YYX$ (desempilhou X e empilhou o YX);

Transições
:: Exemplo 3



$\delta(A, a, \varepsilon) = (B, ZZ)$

Supondo:

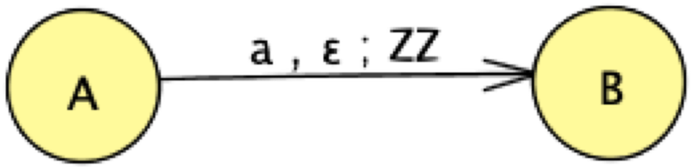
Palavra de entrada $\omega = abcba$;

Estado atual $= A$;

Pilha atual $= XYYX$

A transição ocorre ou não?

Transições
:: Exemplo 3



$\delta(A, a, \varepsilon) = (B, ZZ)$

Supondo:

Palavra de entrada $\omega = abcba$;

Estado atual $= A$;

Pilha atual $= XYYX$

Neste caso a transição ocorre

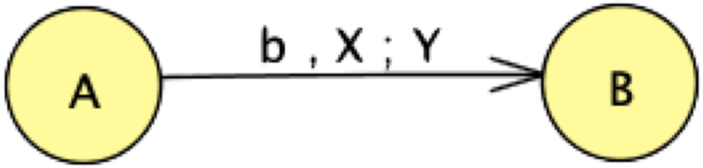
Após a transição...

Palavra de entrada $\omega = bcba$ (consumiu o a);

Estado atual $= B$;

Pilha atual $= \mathrm{ZZ}XYYX$ (desempilhou ε e empilhou o ZZ);

Transições :: Exemplo 4



$\delta(A, b, X) = (B, Y)$

Supondo:

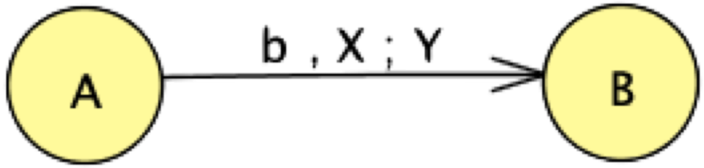
Palavra de entrada $\omega = abcba$;

Estado atual $= A$;

Pilha atual $= XYYX$

A transição ocorre ou não?

Transições
:: Exemplo 4



$$\delta(A, b, X) = (B, Y)$$

Supondo:

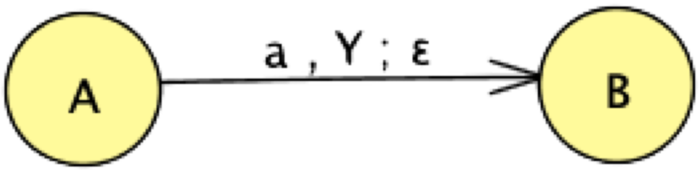
Palavra de entrada $\omega = abcba$;

Estado atual $= A$;

Pilha atual $= XYYX$

Neste caso a transição **não** ocorre porque o símbolo na entrada é a e não b

Transições :: Exemplo 5



$$\delta(A, a, X) = (B, \varepsilon)$$

Supondo:

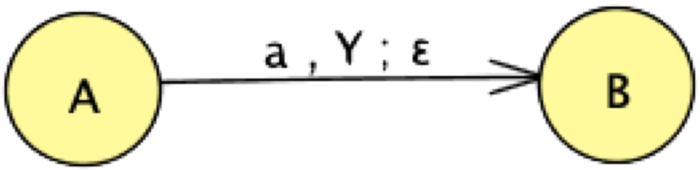
Palavra de entrada $\omega = abcba$;

Estado atual $= A$;

Pilha atual $= XYYX$

A transição ocorre ou não?

Transições
:: Exemplo 5



$\delta(A, a, Y) = (B, \varepsilon)$

Supondo:

Palavra de entrada $\omega = abcba$;

Estado atual $= A$;

Pilha atual $= XYYX$

Neste caso a transição **não** ocorre porque até que é possível consumir o a da entrada, mas não dá para desempilhar Y

Casos de Uso

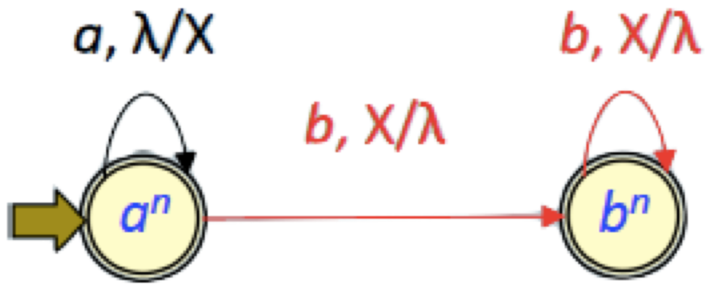
O caso de uso mais comum de APs é utilizar a pilha como um contador de símbolos

Em uma primeira etapa, a pilha é populada para registrar o número de ocorrências de um determinado símbolo de entrada

Posteriormente, a pilha é esvaziada para controlar o número de ocorrências de outro símbolo de entrada

Casos de Uso
:: Exemplo 1

$L_1 = \{a^n b^n : n \geq 0\}$

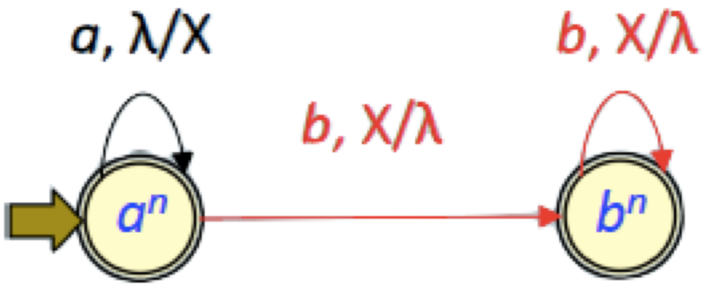


A primeira transição **empilha** um símbolo X para cada símbolo a lido na entrada, i.e., $(a^n, a, \varepsilon) \rightarrow (b^n, X)$

As demais transições **desempilham** um símbolo X para cada símbolo b lido na entrada , p.ex., $(b^n, b, X) \rightarrow (b^n, \varepsilon)$

Casos de Uso
:: Exemplo 1

$L_1 = \{a^n b^n : n \geq 0\}$



$\omega = aabb$

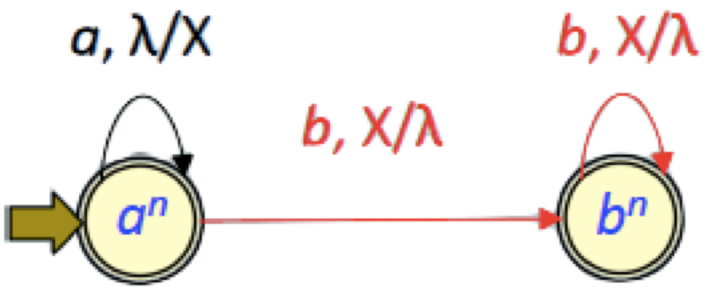
Símbolo lido $= \varepsilon$

Estado atual $= a^n$

Pilha atual $= \varepsilon$

Casos de Uso
:: Exemplo 1

$L_1 = \{a^n b^n : n \geq 0\}$



$\omega = abb$

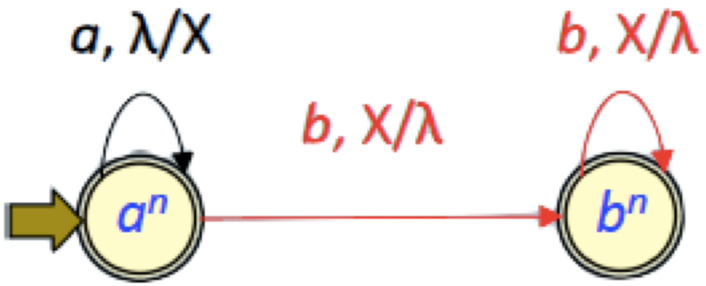
Símbolo lido $= a$

Estado atual $= a^n$

Pilha atual $= X$

Casos de Uso
:: Exemplo 1

$L_1 = \{a^n b^n : n \geq 0\}$



$\omega = bb$

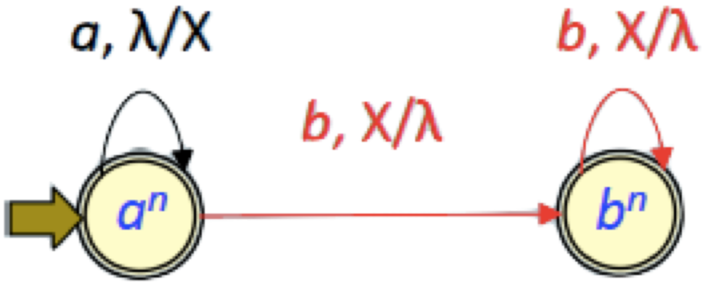
Símbolo lido $= a$

Estado atual $= a^n$

Pilha atual $= XX$

Casos de Uso
:: Exemplo 1

$L_1 = \{a^n b^n : n \geq 0\}$



$\omega = b$

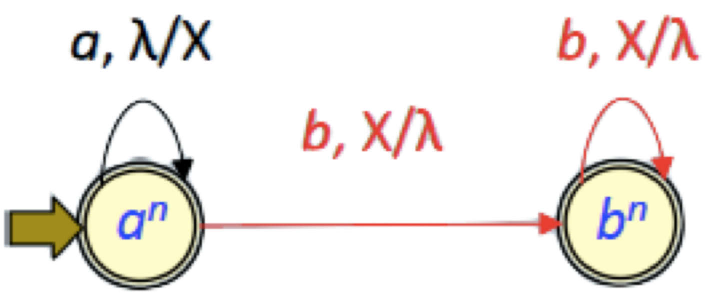
Símbolo lido $= b$

Estado atual $= b^n$

Pilha atual $= X$

Casos de Uso :: Exemplo 1

$L_1 = \{a^n b^n : n \geq 0\}$



$\omega = \varepsilon$

Símbolo lido $= b$

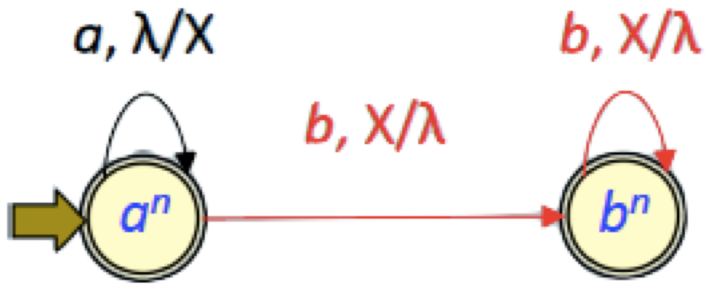
Estado atual $= b^n$

Pilha atual $= \varepsilon$

Resultado: (i) cadeia consumida; (ii) Estado de aceitação; (iii) Pilha vazia; e (iv) cadeia reconhecida

Casos de Uso
:: Exemplo 2

$L_1 = \{a^n b^n : n \geq 0\}$



$\omega = aaabb$

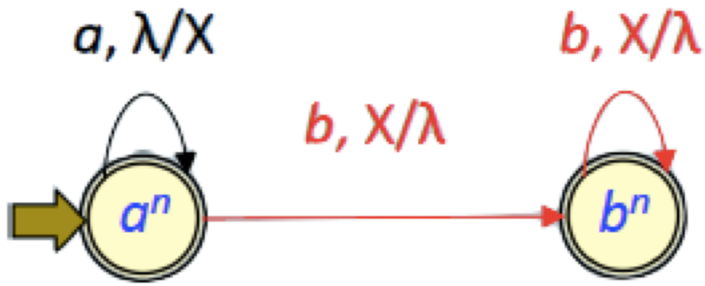
Símbolo lido = ε

Estado atual = a^n

Pilha atual = ε

Casos de Uso
:: Exemplo 2

$L_1 = \{a^n b^n : n \geq 0\}$



$\omega = aabb$

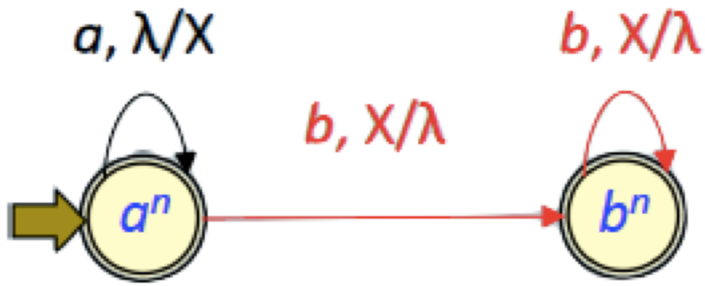
Símbolo lido = a

Estado atual = a^n

Pilha atual = X

Casos de Uso
:: Exemplo 2

$L_1 = \{a^n b^n : n \geq 0\}$



$\omega = abb$

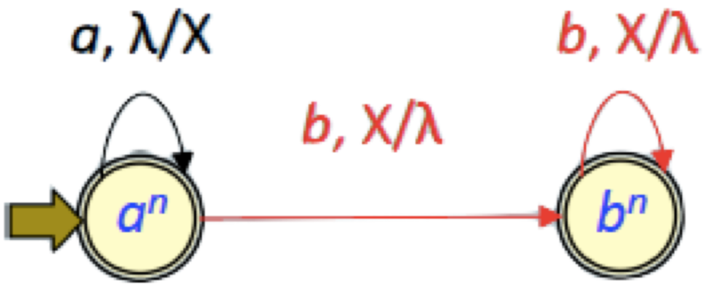
Símbolo lido = a

Estado atual = a^n

Pilha atual = XX

Casos de Uso
:: Exemplo 2

$L_1 = \{a^n b^n : n \geq 0\}$



$\omega = bb$

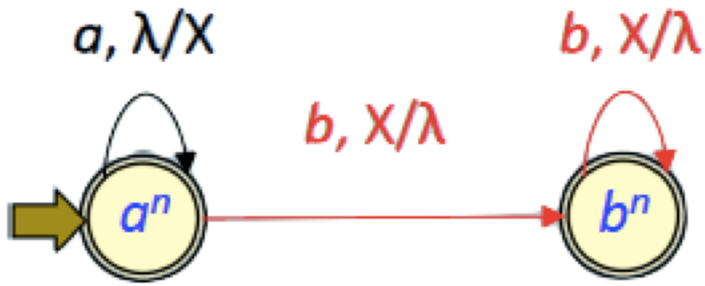
Símbolo lido $=$ a

Estado atual $= a^n$

Pilha atual $= XXX$

Casos de Uso
:: Exemplo 2

$L_1 = \{a^n b^n : n \geq 0\}$



$\omega = b$

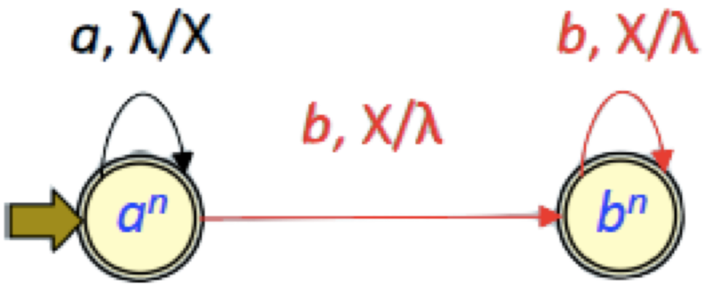
Símbolo lido $= b$

Estado atual $= b^n$

Pilha atual $= XX$

Casos de Uso
:: Exemplo 2

$L_1 = \{a^n b^n : n \geq 0\}$



$\omega = \varepsilon$

Símbolo lido $= b$

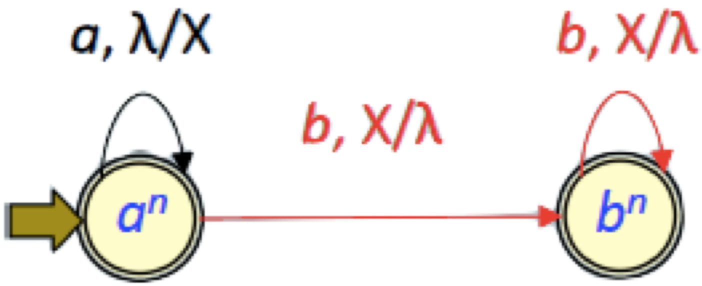
Estado atual $= b^n$

Pilha atual $= X$

Resultado: (i) cadeia consumida; (ii) Estado de aceitação; (iii) Pilha **não** vazia; e (iv) cadeia **não** reconhecida

Casos de Uso :: Exemplo 3

$L_1 = \{a^n b^n : n \geq 0\}$



$\omega = aabbb$

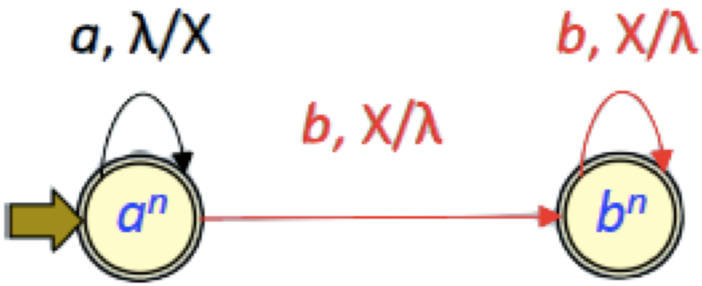
Símbolo lido $= \varepsilon$

Estado atual $= a^n$

Pilha atual $= \varepsilon$

Casos de Uso
:: Exemplo 3

$L_1 = \{a^n b^n : n \geq 0\}$



$\omega = abbb$

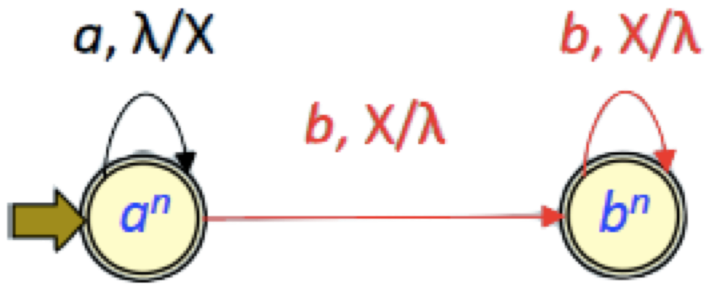
Símbolo lido = a

Estado atual = a^n

Pilha atual = X

Casos de Uso
:: Exemplo 3

$L_1 = \{a^n b^n : n \geq 0\}$



$\omega = bbb$

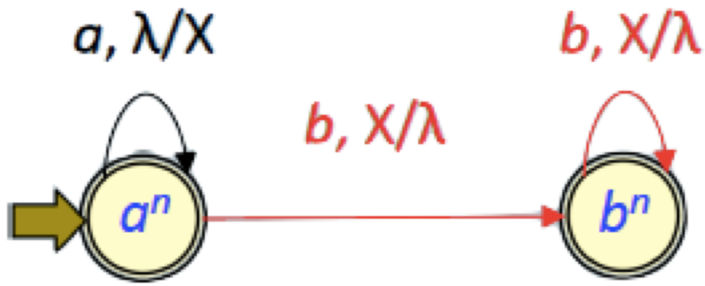
Símbolo lido = a

Estado atual = a^n

Pilha atual = XX

Casos de Uso
:: Exemplo 3

$L_1 = \{a^n b^n : n \geq 0\}$



$\omega = bb$

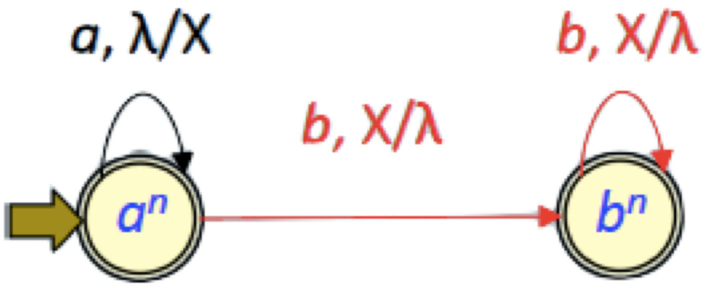
Símbolo lido $= b$

Estado atual $= b^n$

Pilha atual $= X$

Casos de Uso :: Exemplo 3

$L_1 = \{a^n b^n : n \geq 0\}$



$\omega = b$

Símbolo lido $= b$

Estado atual $= b^n$

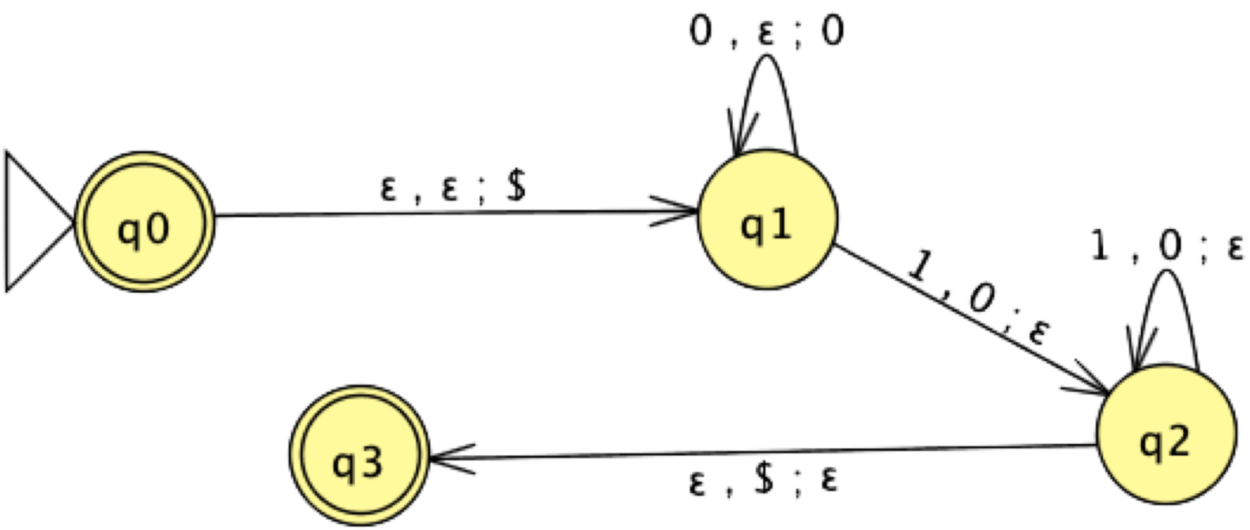
Pilha atual $= \varepsilon$

Resultado: (i) cadeia **não** consumida; (ii) Estado de aceitação; (iii) Pilha vazia; e (iv) cadeia **não** reconhecida

Exemplos no JFLAP

Exemplo 1

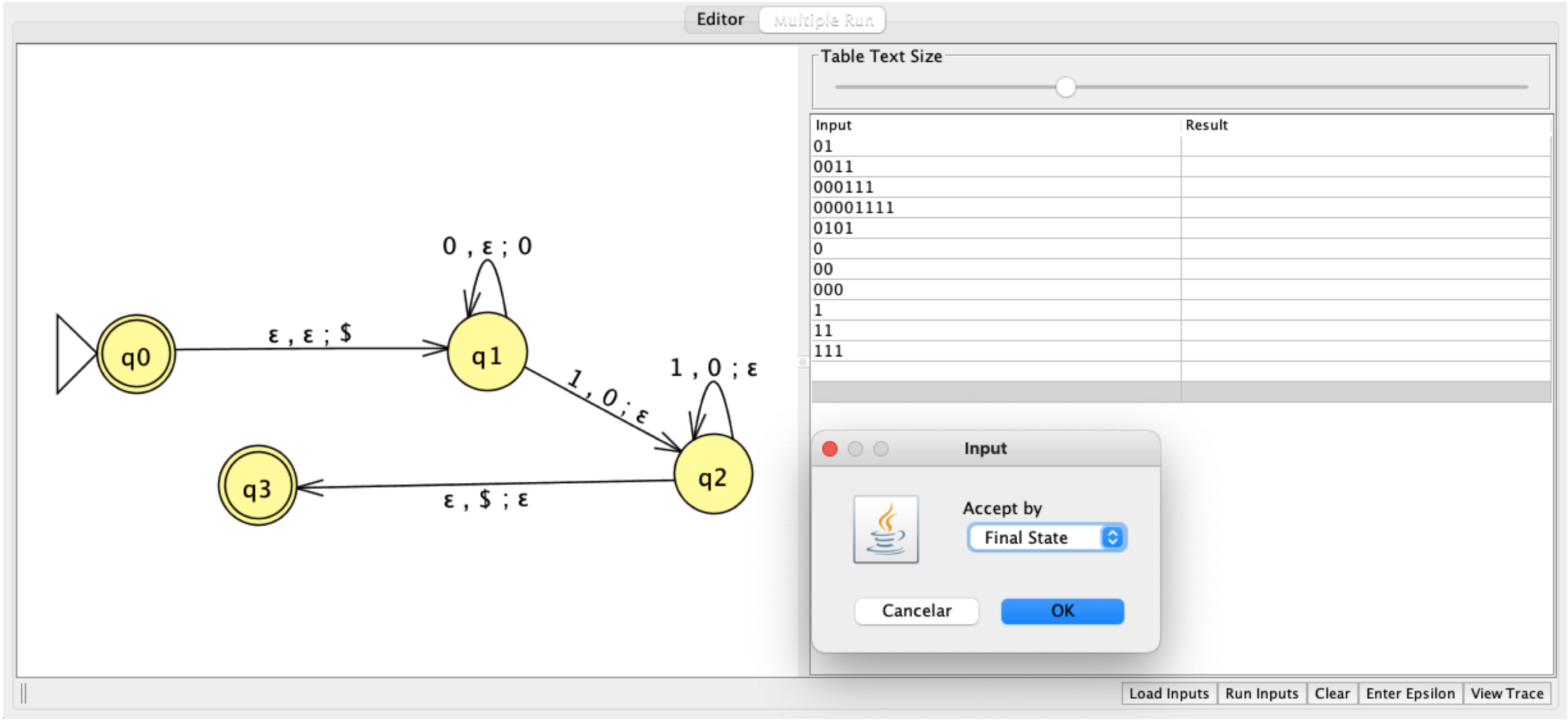
$L_1 = \{0^n 1^n : n \geq 0\}$



Inicialmente, inclui o símbolo "$". A cada "0" lido, desempilha nada e empilha "0". A cada "1" lido, desempinha um "0" e empilha nada. No final, tem que ler o símbolo "$" (significa pilha vazia)

Exemplo 1

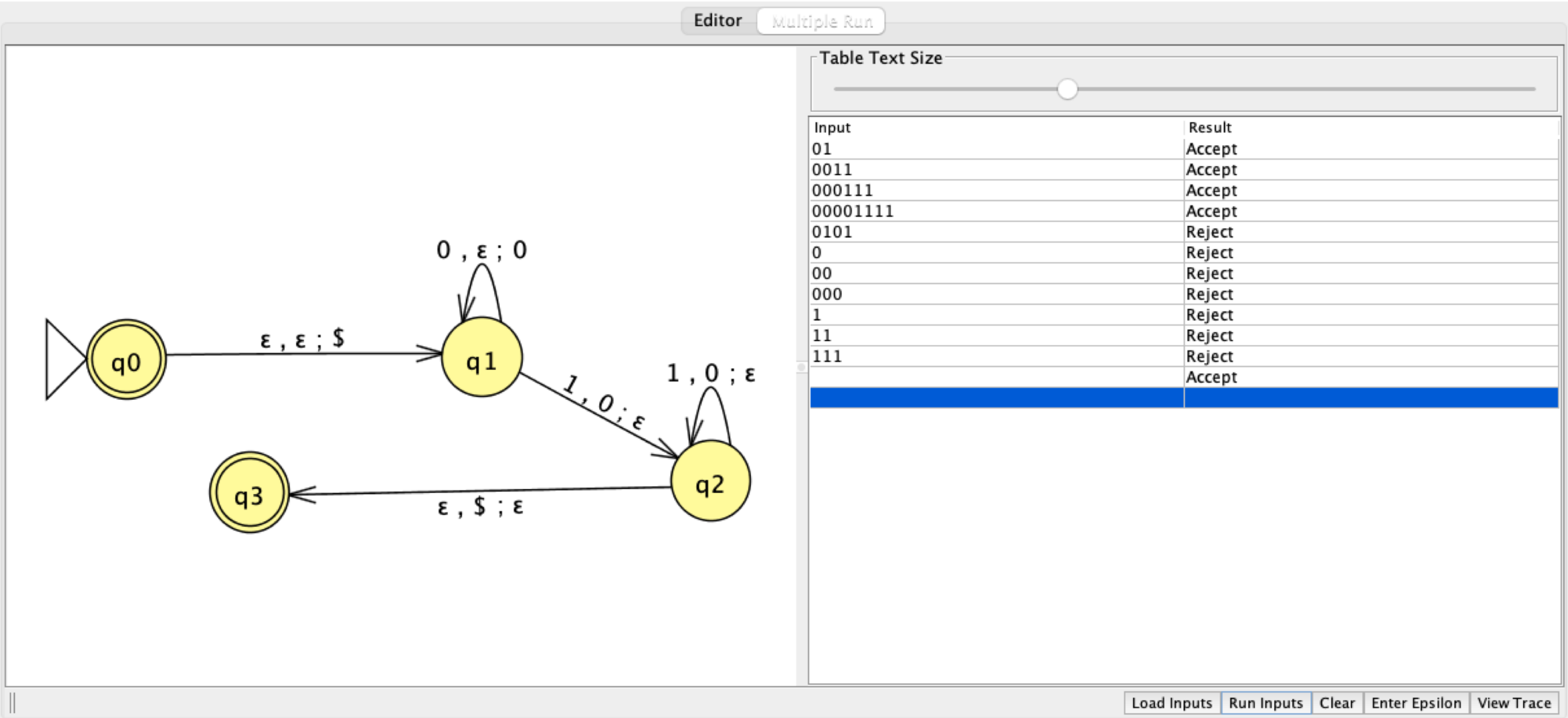
$$L_1 = \{0^n 1^n : n \geq 0\}$$



Aceitação ao chegar ao estado final (aceitação)

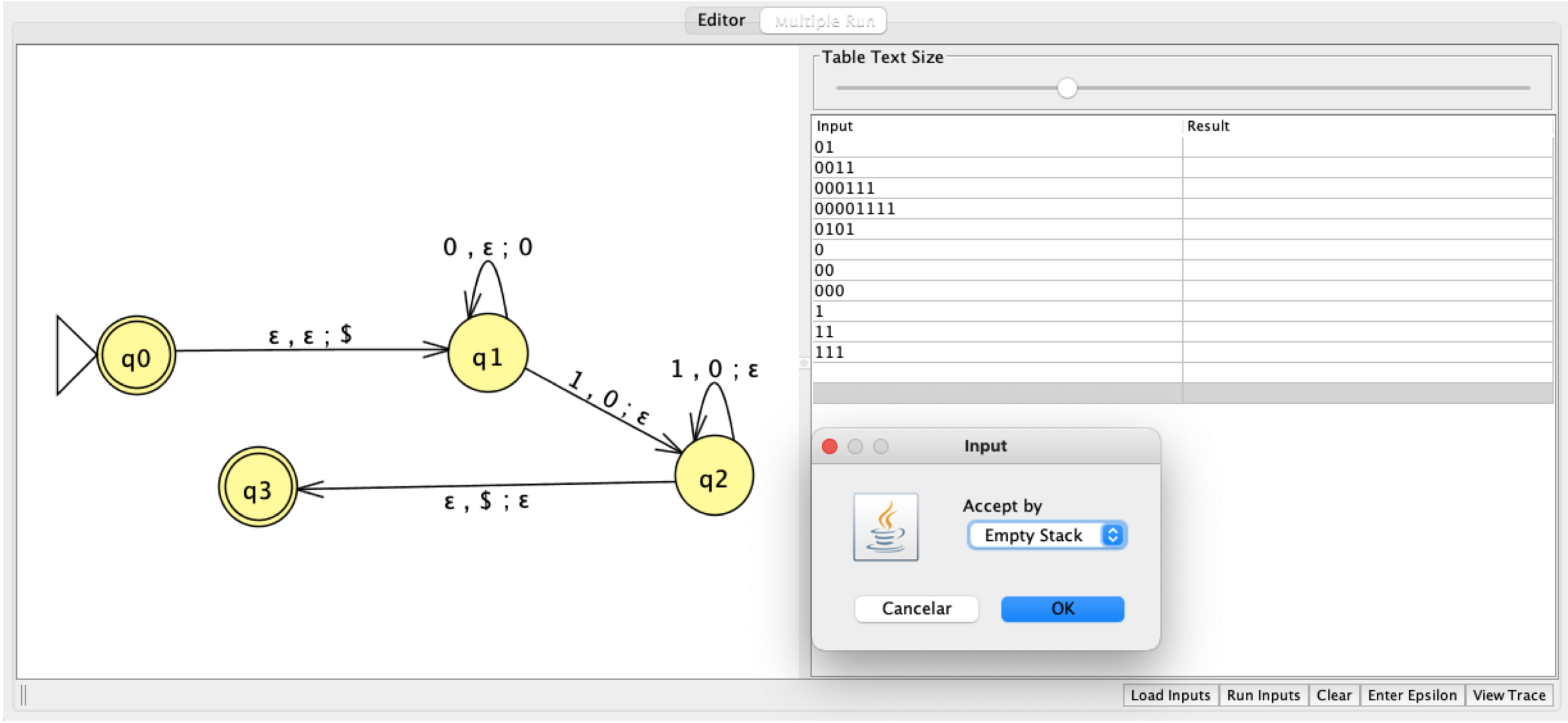
Exemplo 1

$$L_1 = \{0^n 1^n : n \geq 0\}$$



Exemplo 1

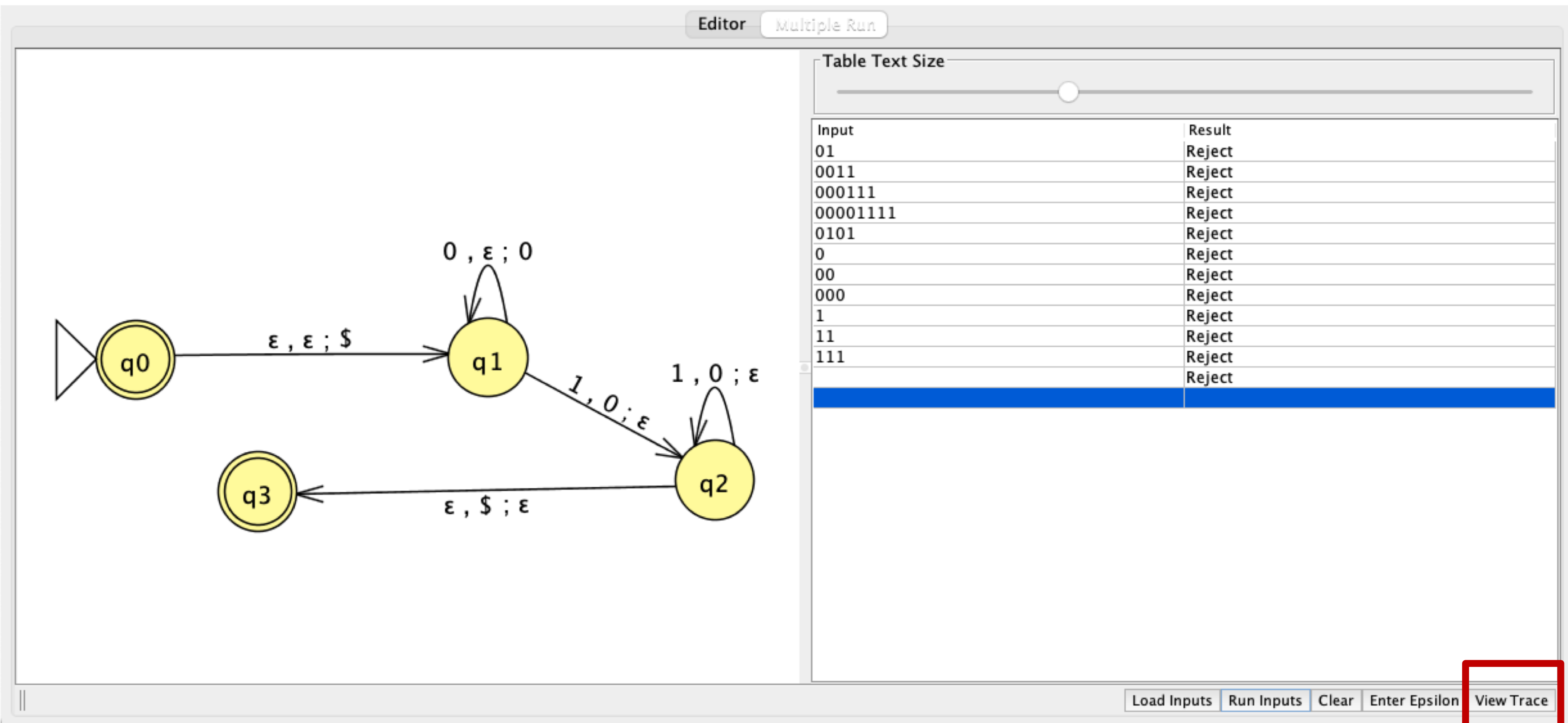
$L_1 = \{0^n 1^n : n \geq 0\}$



Aceitação ao atingir pilha vazia

Exemplo 1

$L_1 = \{0^n 1^n : n \geq 0\}$



Neste caso, rejeitou todas! É possível também usar "**View Trace**"

Exemplo 1

$L_1 = \{0^n 1^n : n \geq 0\}$

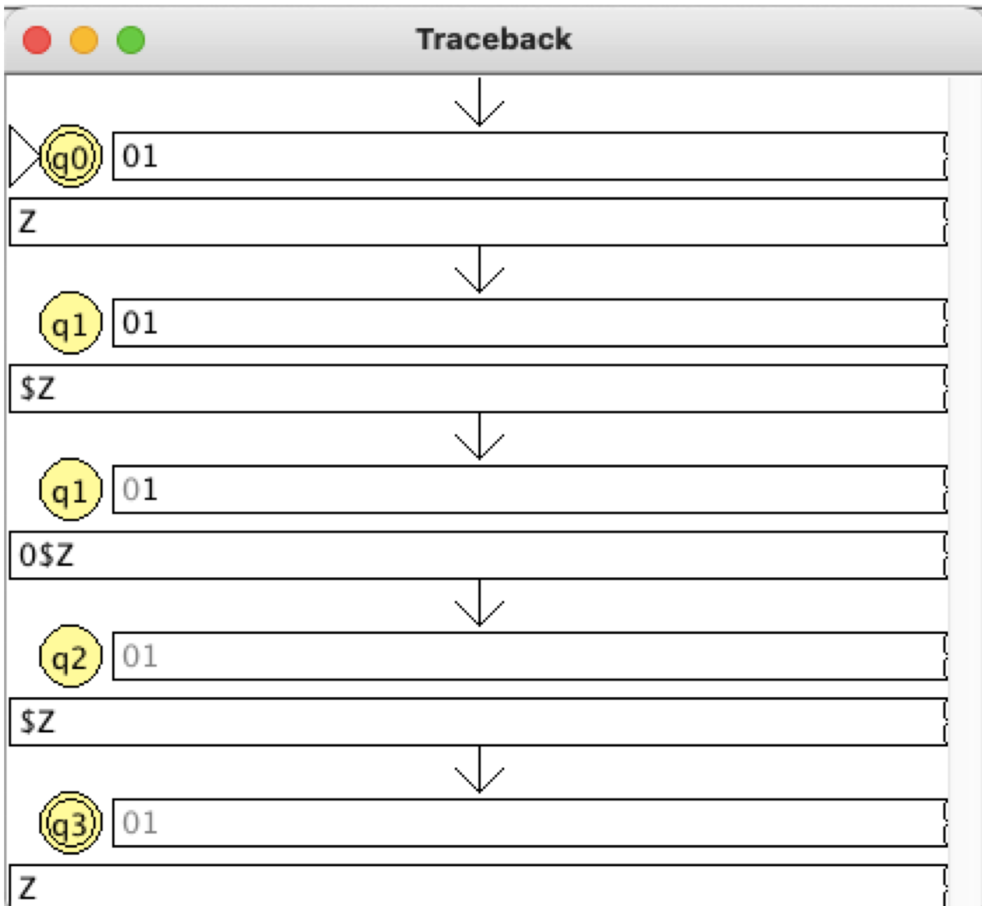
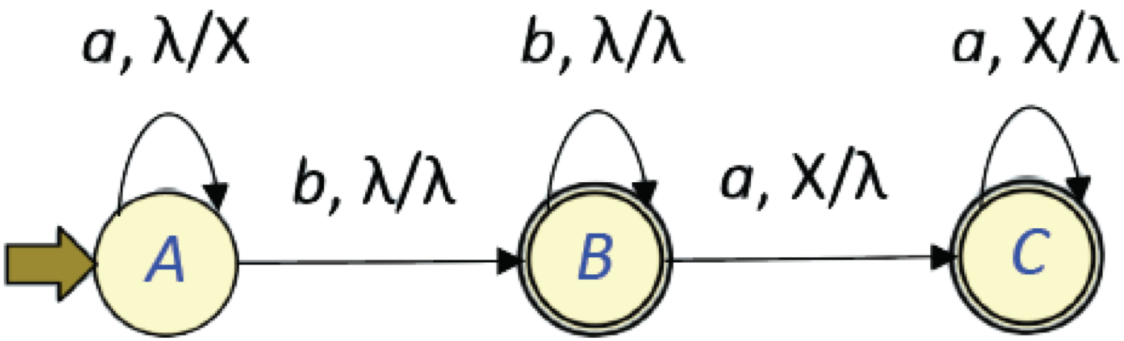


Table Text Size

Input	Result
01	Accept
0011	Accept
000111	Accept
00001111	Accept
0101	Reject
0	Reject
00	Reject
000	Reject
1	Reject
11	Reject
111	Reject
	Accept

Exemplo 2

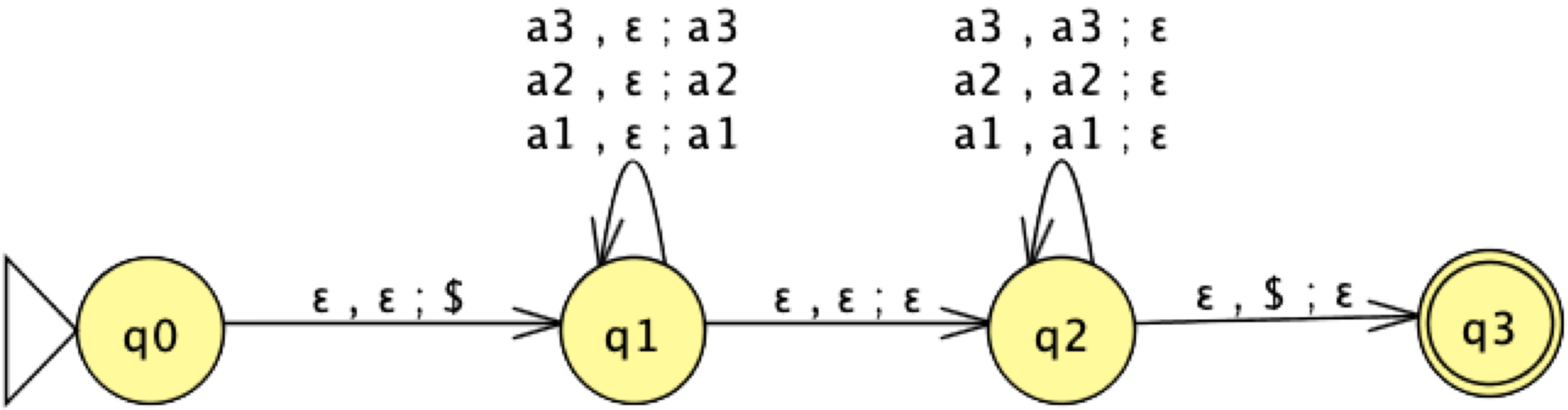
$L_2 = \{a^n b^m a^n : n \geq 0, m \geq 0\}$



O "contador" implementado pela pilha **não precisa ser utilizado imediatamente,** mas a contagem pode ser necessária em um ponto mais adiante

Exemplo 3

$L_3 = \{\omega\omega^R : \omega \in \{a1, a2, a3\}\}$

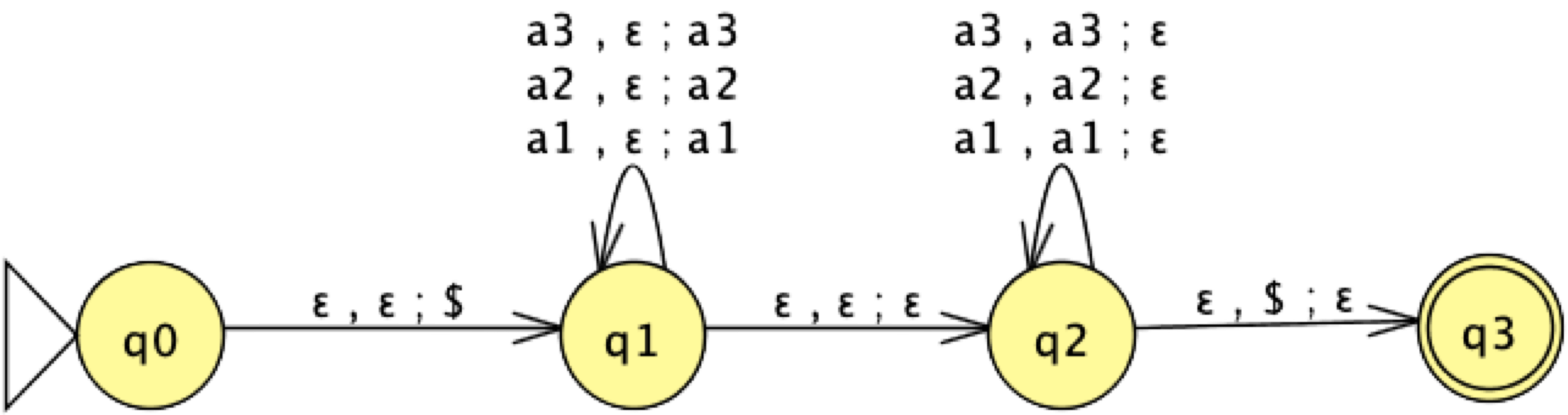


Apesar de serem geradas várias cópias das máquinas, o foco será só na cópia que vai chegar ao final com êxito

$

Exemplo 3

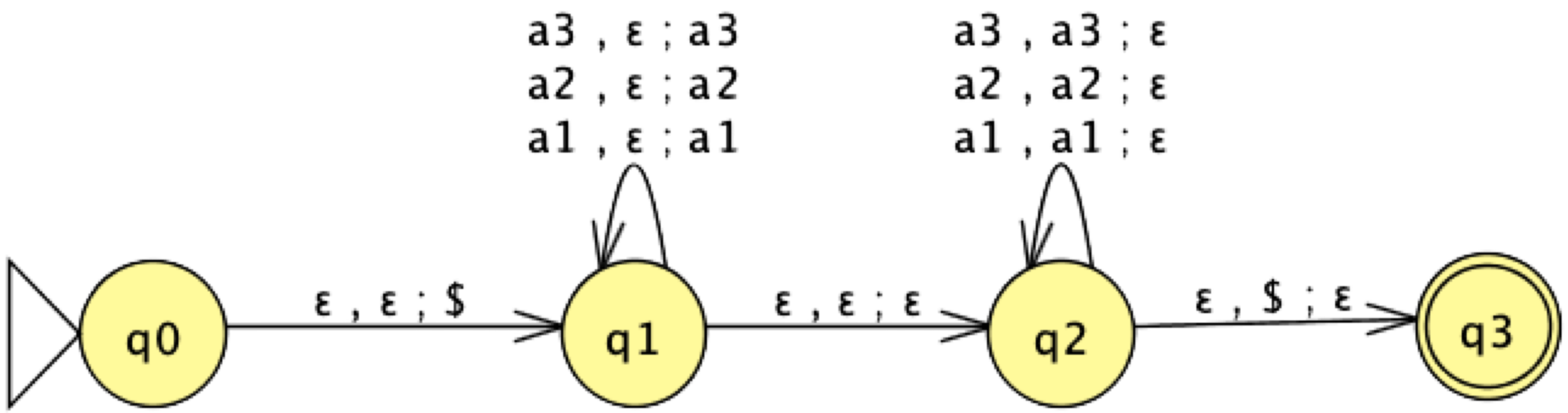
$$L_3 = \{\omega\omega^R : \omega \in \{a1, a2, a3\}$$



a1
$

Exemplo 3

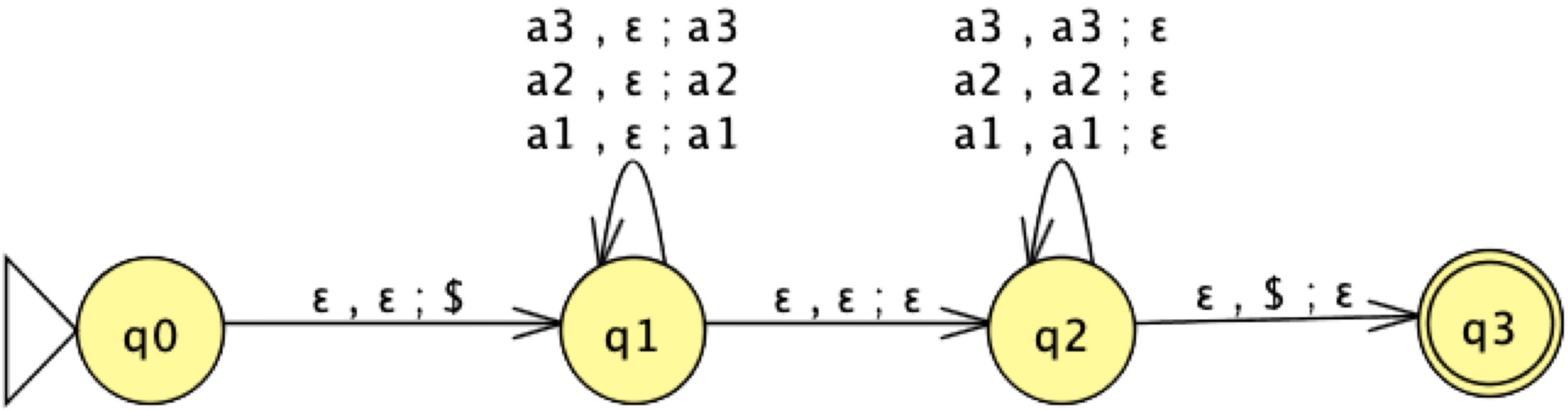
$L_3 = \{\omega\omega^R : \omega \in \{a1, a2, a3\}\}$

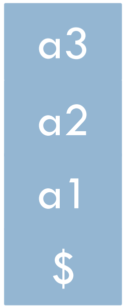


a2
a1
$

Exemplo 3

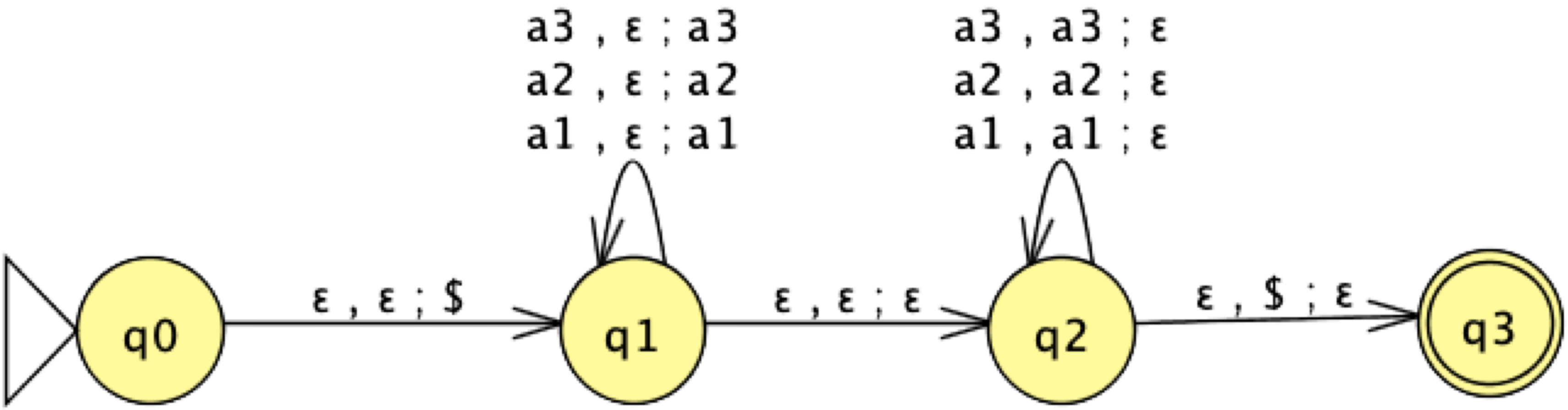
$$L_3 = \{\omega\omega^R : \omega \in \{a1, a2, a3\}\}$$





Exemplo 3

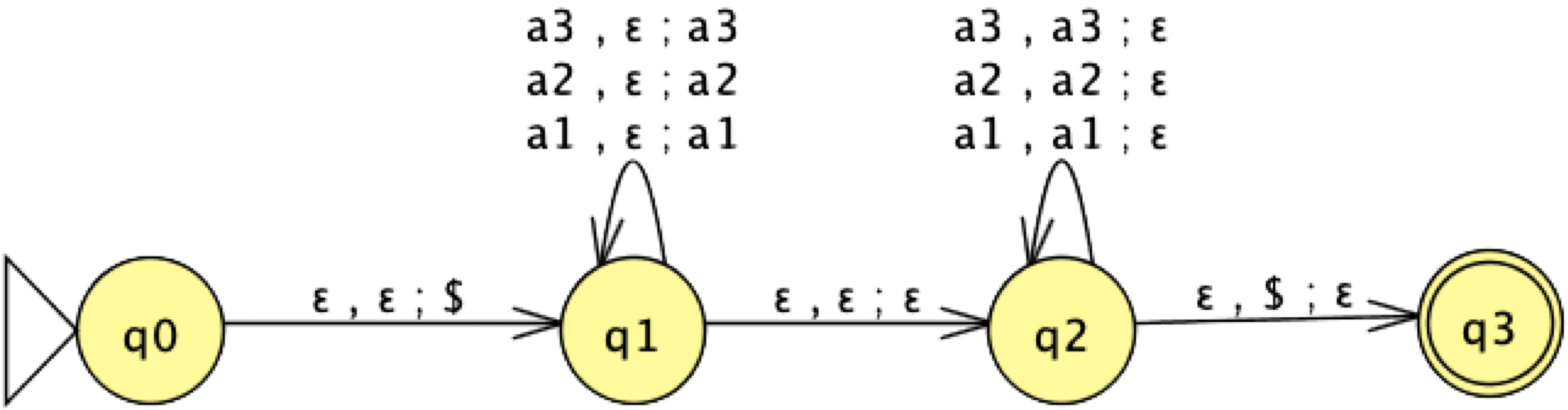
$L_3 = \{\omega\omega^R : \omega \in \{a1, a2, a3\}$



a2
a1
$

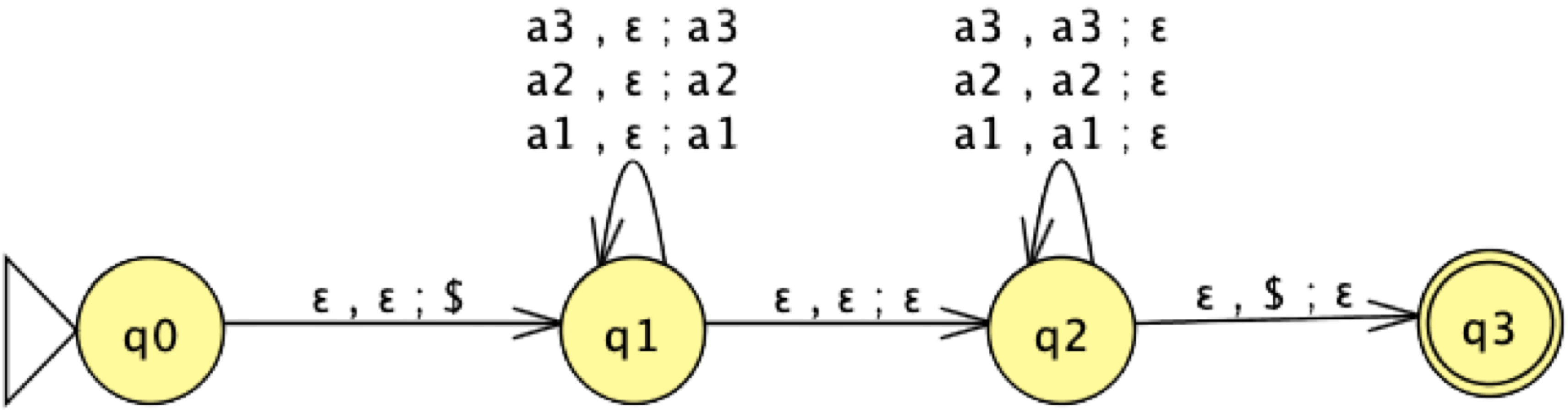
Exemplo 3

$L_3 = \{\omega\omega^R : \omega \in \{a1, a2, a3\}$



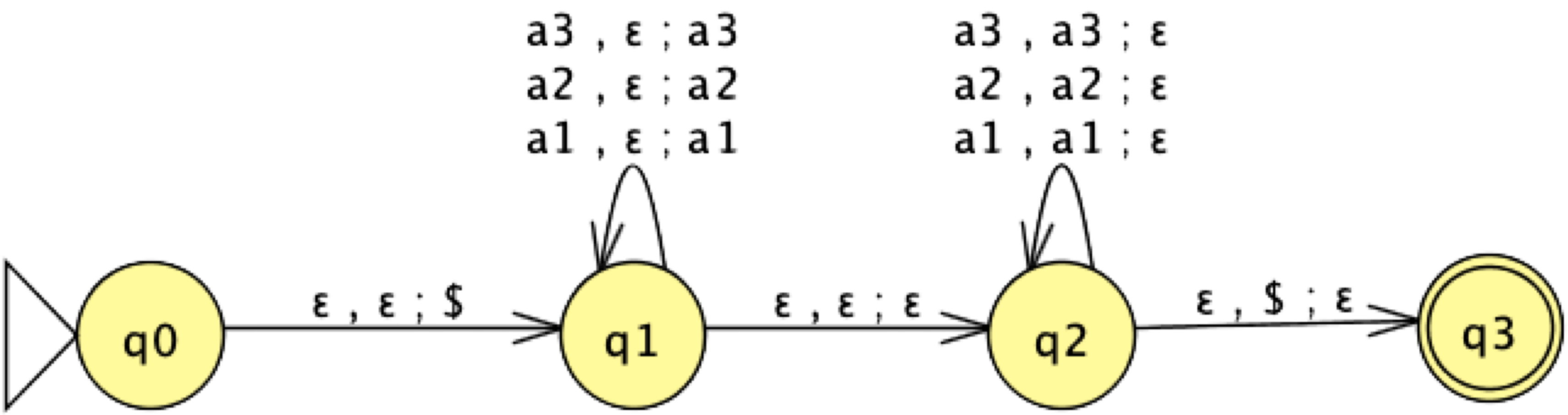
Exemplo 3

$L_3 = \{\omega\omega^R : \omega \in \{a1, a2, a3\}$



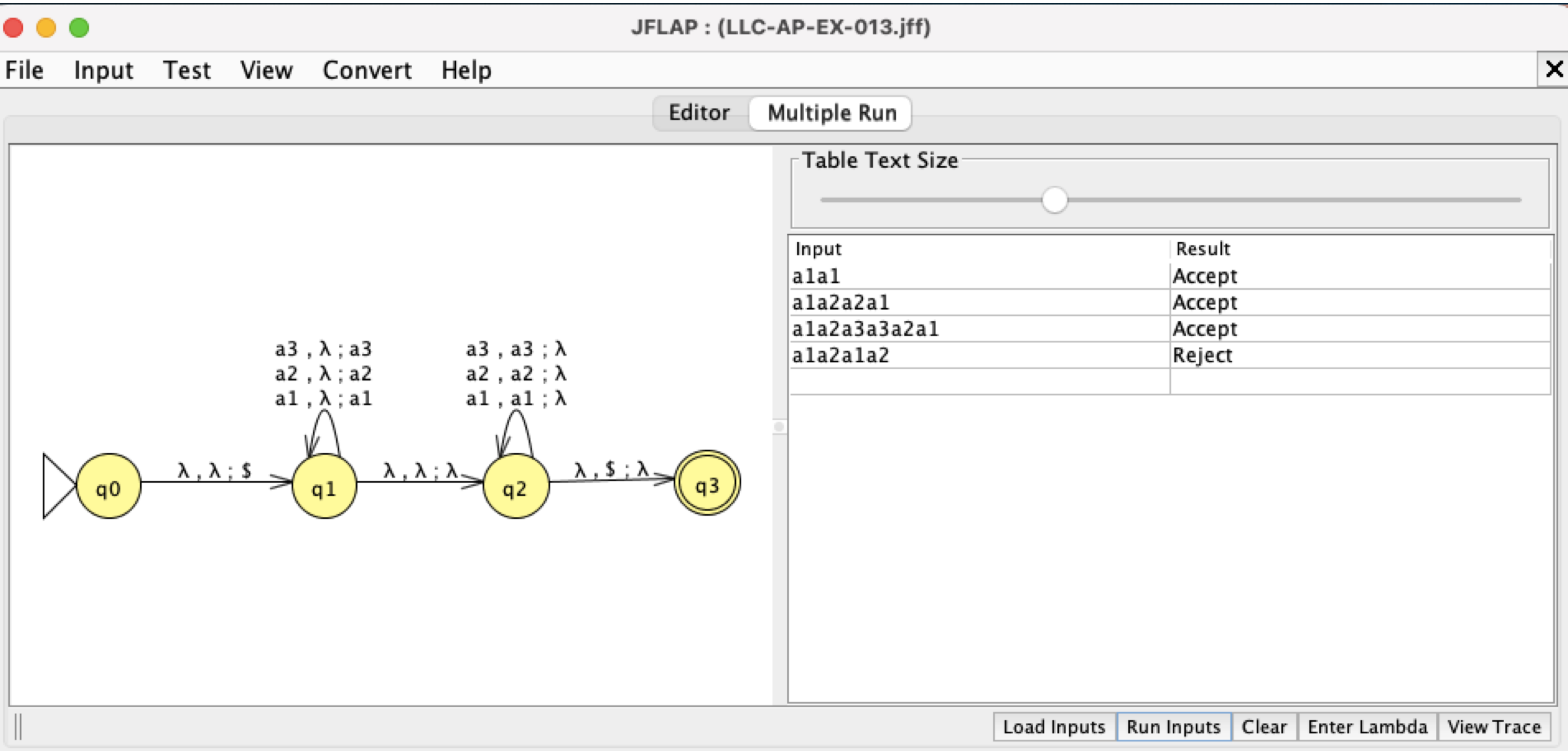
Exemplo 3

$$L_3 = \{\omega\omega^R : \omega \in \{a1, a2, a3\}\}$$



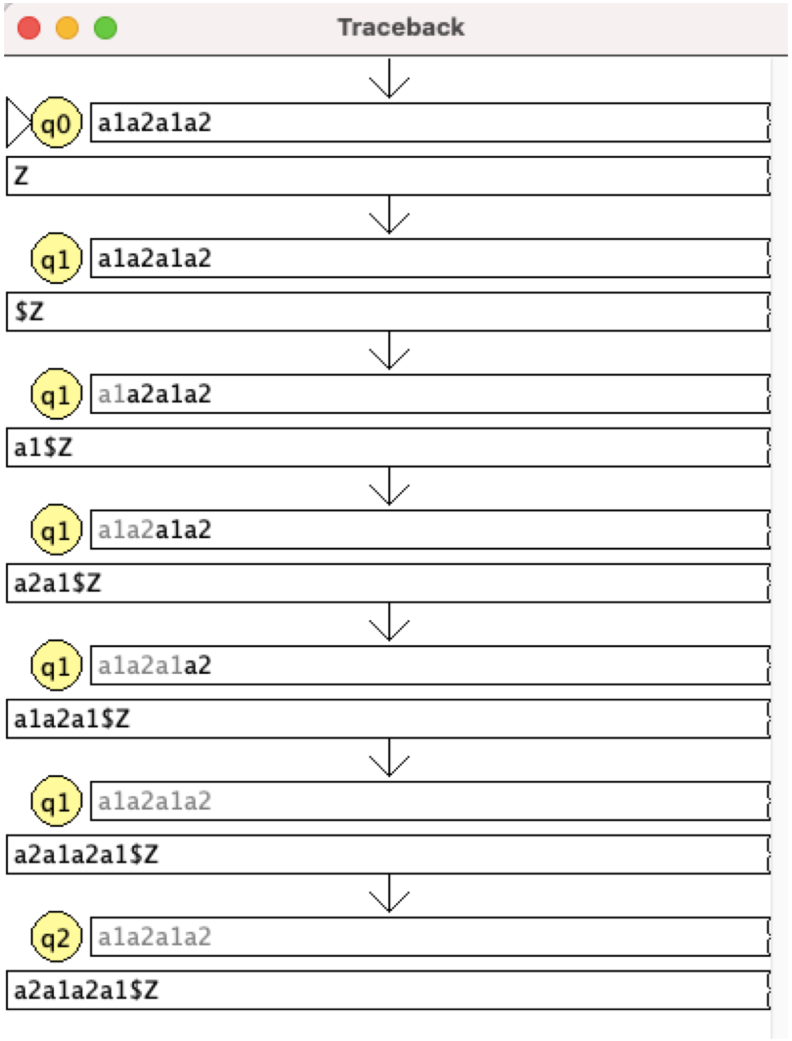
Estado de aceitação & pilha vazia = cadeia aceita!

Exemplo 3



JFLAP : (LLC-AP-EX-013.jff)
File
Input
Test
View
Convert
Help
Editor
Multiple Run
a3 , λ ; a3
a2 , λ ; a2
a1 , λ ; a1
a3 , a3 ; λ
a2 , a2 ; λ
a1 , a1 ; λ
λ , λ ; $
λ , λ ; λ
λ , $; λ
q0
q1
q2
q3
Table Text Size
Input
Result
a1a1
Accept
a1a2a2a1
Accept
a1a2a3a3a2a1
Accept
a1a2a1a2
Reject
Load Inputs
Run Inputs
Clear
Enter Lambda
View Trace

Exemplo 3 (APND)



Autômato com Pilha é o meu precioso!!!