

# Linguagens Formais e Autômatos (LFA)

Aula de 25/09/2013

**Minimização de Autômatos**

## O que é um autômato mínimo?

Um autômato determinístico

que usa o menor número possível de estados.

Autômatos mínimos servem de modelo para programas de decisão (S/N) como o menor número de passos.

## Sob outra perspectiva

Dada uma linguagem regular  $L_{Reg}$  e um autômato  $AF$  que a reconhece,

$AF$  é o autômato mínimo para  $L_{Reg}$  se:

1.  $AF$  é determinístico e tem um número  $Q$  de estados; e
2. Não existe nenhum outro  $AF$  determinístico com  $Q'$  estados e que reconhece  $L_{Reg}$ , sendo  $Q' < Q$ .

"Minimizar" um autômato finito  $AF$  é, portanto, reduzir as transições de estados que sejam equivalentes uma às outras.

## Método de Minimização por Discriminações Sucessivas

A ideia geral:

Seja uma Linguagem Regular  $L$  e um autômato finito determinístico AFD que aceita todas e somente estas cadeias.

Tomar o conjunto completo de estados  $Q$  de AFD e particioná-lo iterativamente em subconjuntos  $Q_1 \dots Q_n$  que sejam distintos entre si.

As partições indistinguíveis finais do processo constituem o conjunto mínimo de estados  $Q_{min}$  distintos necessários para reconhecer a mesma linguagem que o autômato não-mínimo reconhece.

O AFD mínimo para  $L$  é construído sobre o conjunto  $Q_{min}$ , especificando-se as transições necessárias para que todas as cadeias de  $L$  sejam reconhecidas.

## Processo iterado de partições:

Como raramente os estados têm transições para TODOS os símbolos do alfabeto, ao início do processo cria-se um estado adicional (inócuo) para consumir os símbolos que importam para a discriminação, mas não para a aceitação da linguagem.

Na primeira iteração do particionamento de estados, há uma grande distinção inicial a fazer: **ESTADOS NÃO FINAIS X ESTADOS FINAIS**.

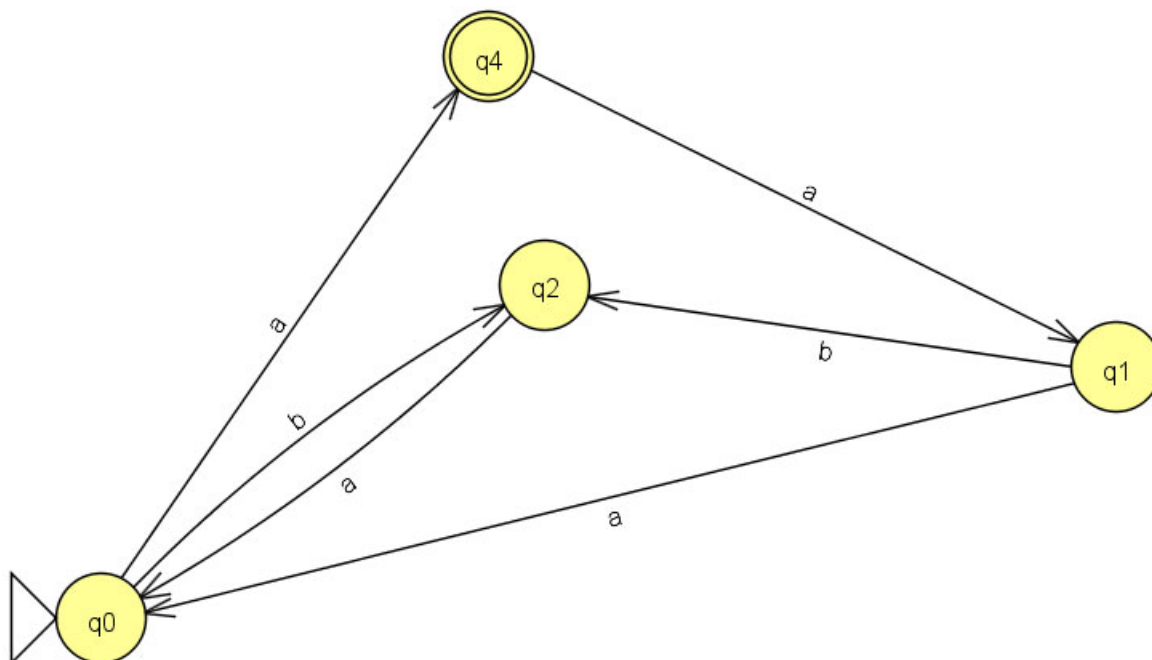
Os grupos da partição feita constituem as folhas da árvore de partições corrente, as partições são disjuntas e sua união resulta no conjunto completo de estados.

As próximas partições tomam por chave cada um dos símbolos do alfabeto do autômato e discrimina se as transições definidas para ele levam a estados da mesma partição ou da outra.

Se todas as transições levam a estados da mesma partição, o símbolo em questão não discrimina os estados. Toma-se o próximo.

As discriminações resultantes subdividem uma partição em duas ou mais alternativas.

## Exemplo do procedimento (JFLAP)

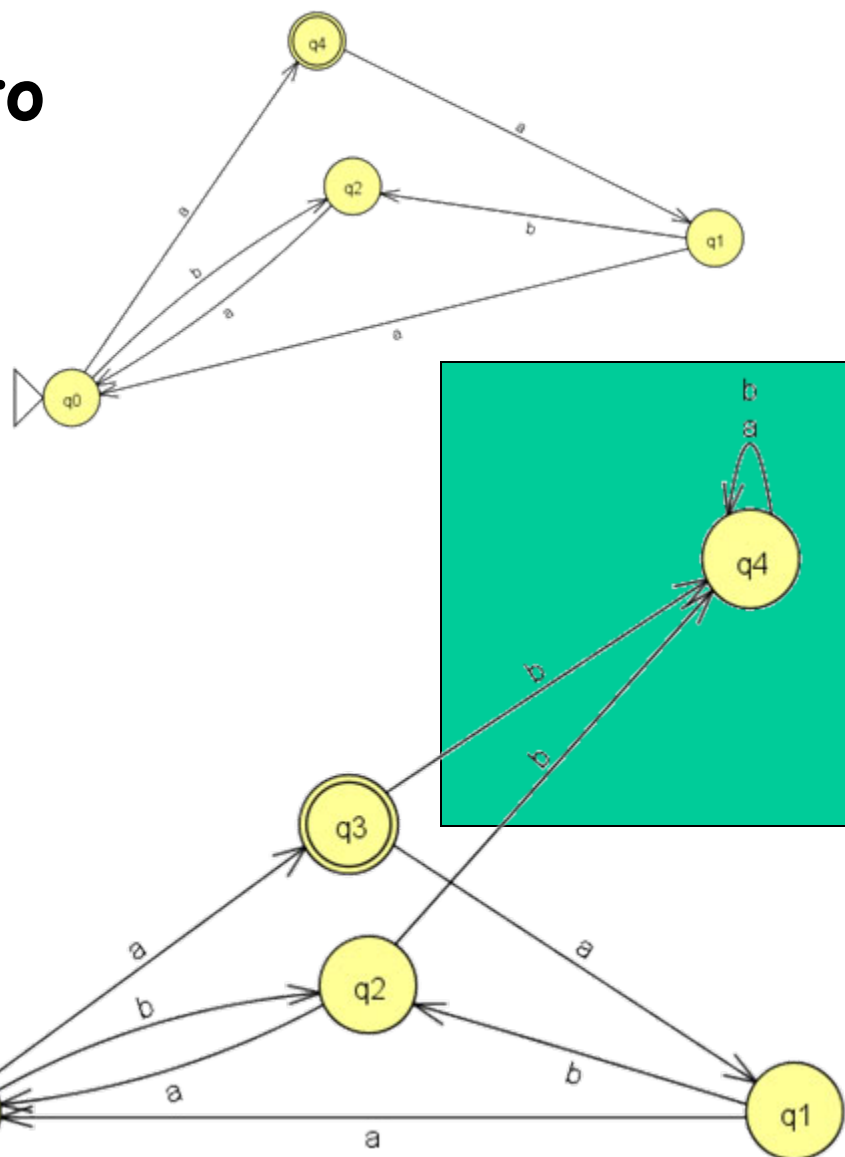


O autômato AFD  
ao lado  
reconhece L.

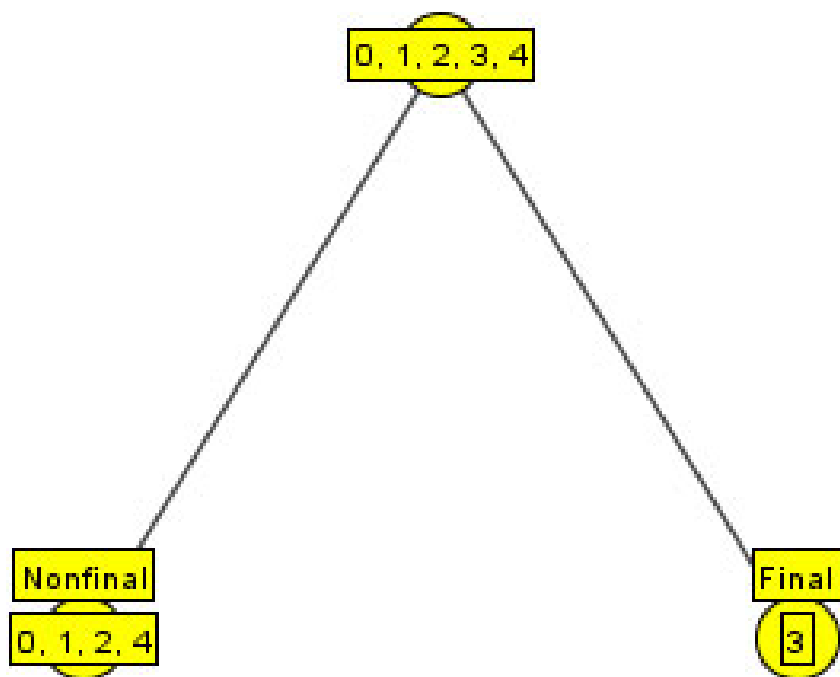
Qual o padrão das  
cadeias de L?

## Exemplo em andamento

Como raramente os estados têm transições para TODOS os símbolos do alfabeto, cria-se um estado adicional (inócuo) para consumir os símbolos que importam para a discriminação, mas não para a aceitação da linguagem.



## Estado Corrente da Partição



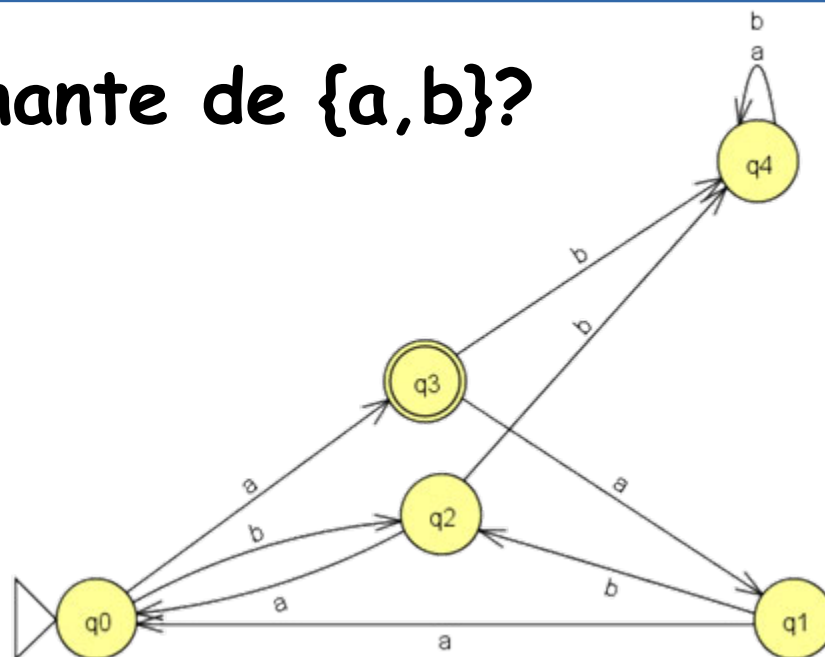
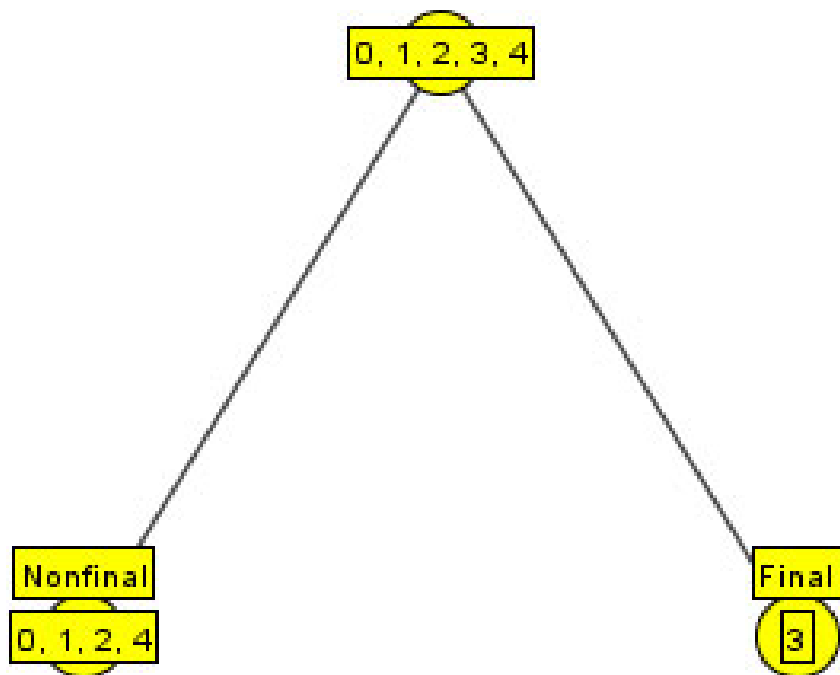
Estados não finais: incluindo o novo estado consumidor  $\{q_0, q_1, q_2, q_4\}$

Estado final:  $\{q_3\}$

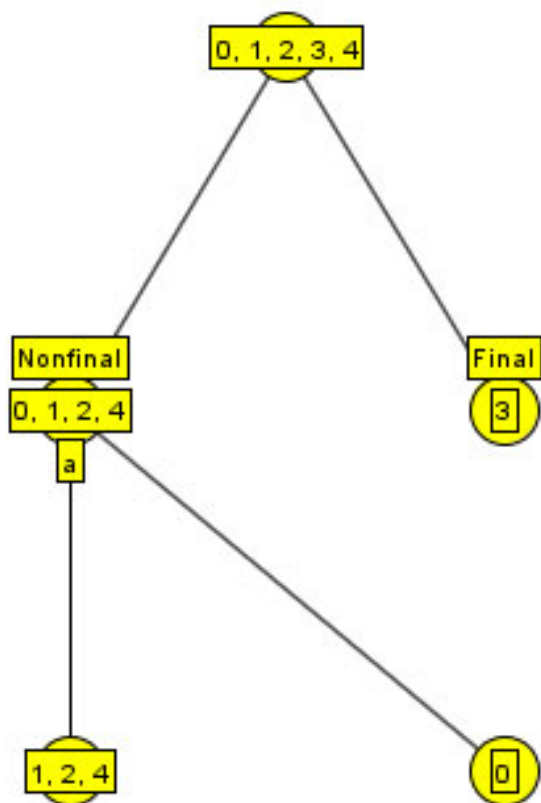
Próxima discriminação?



Qual o símbolo discriminante de  $\{a, b\}$ ?



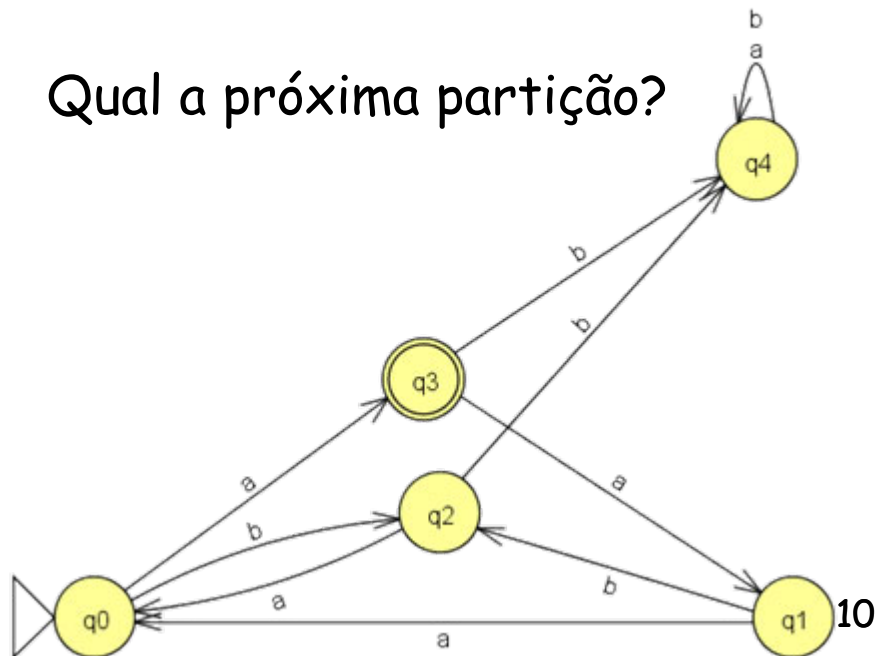
## Partição resultante:



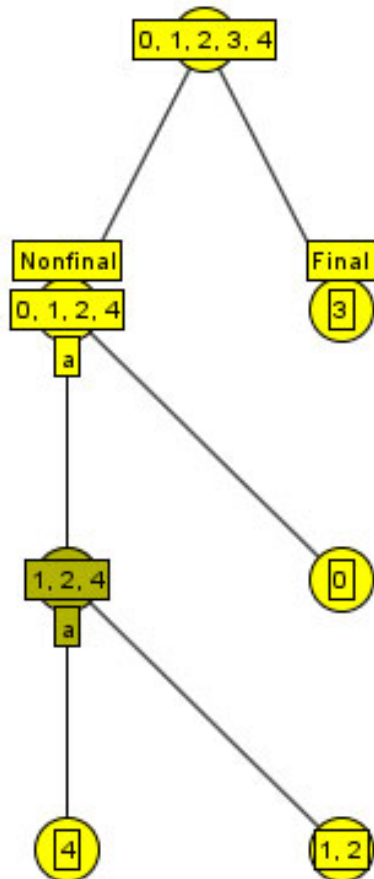
"a" discrimina  $\{q_1, q_2, q_4\}$  de  $\{q_0\}$

"b" discrimina o nó de não-finais?

Qual a próxima partição?



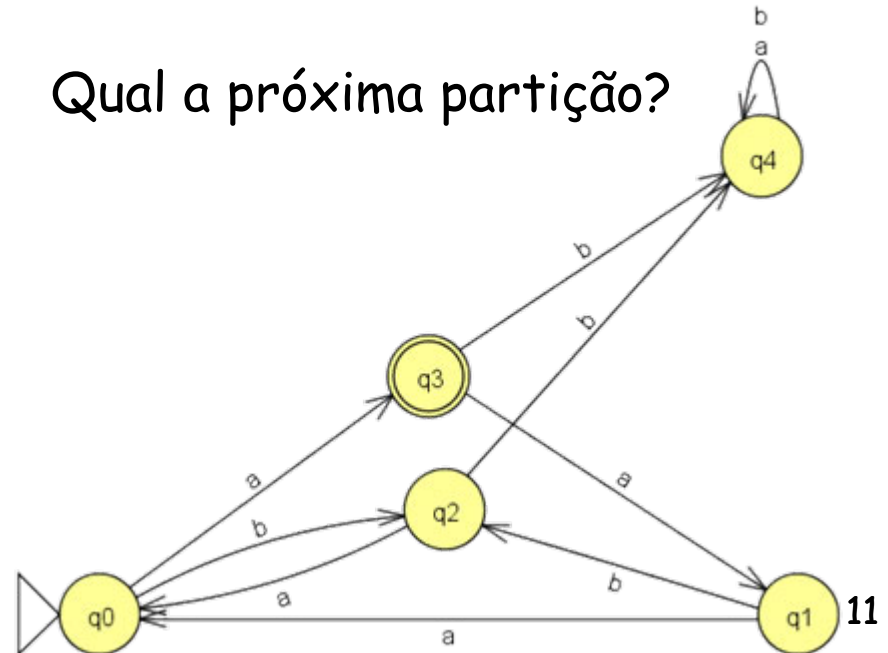
## Partição resultante:



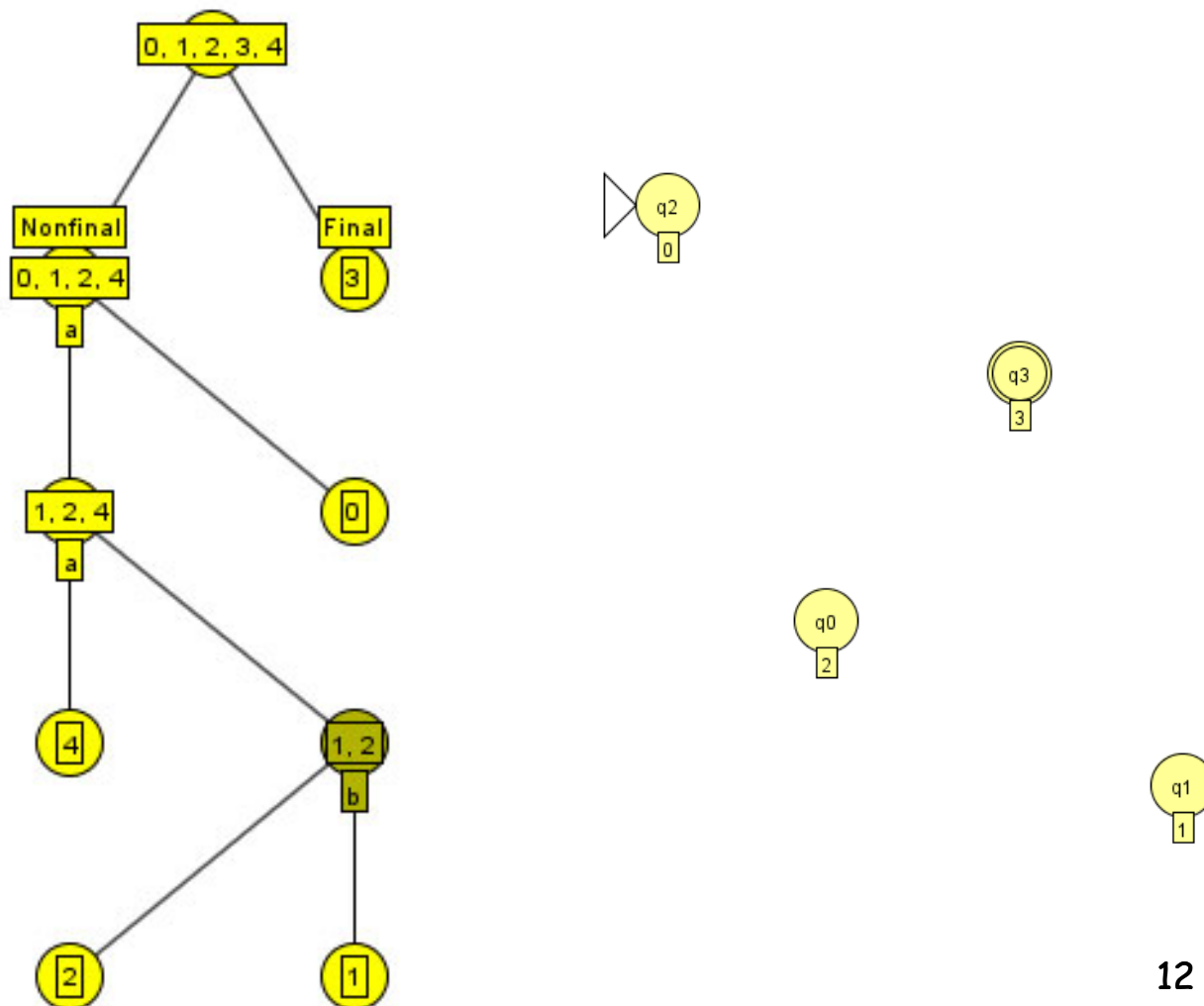
"a" discrimina  $\{q_1, q_2, q_4\}$  de  $\{q_0\}$

"b" discrimina o nó de não-finais?

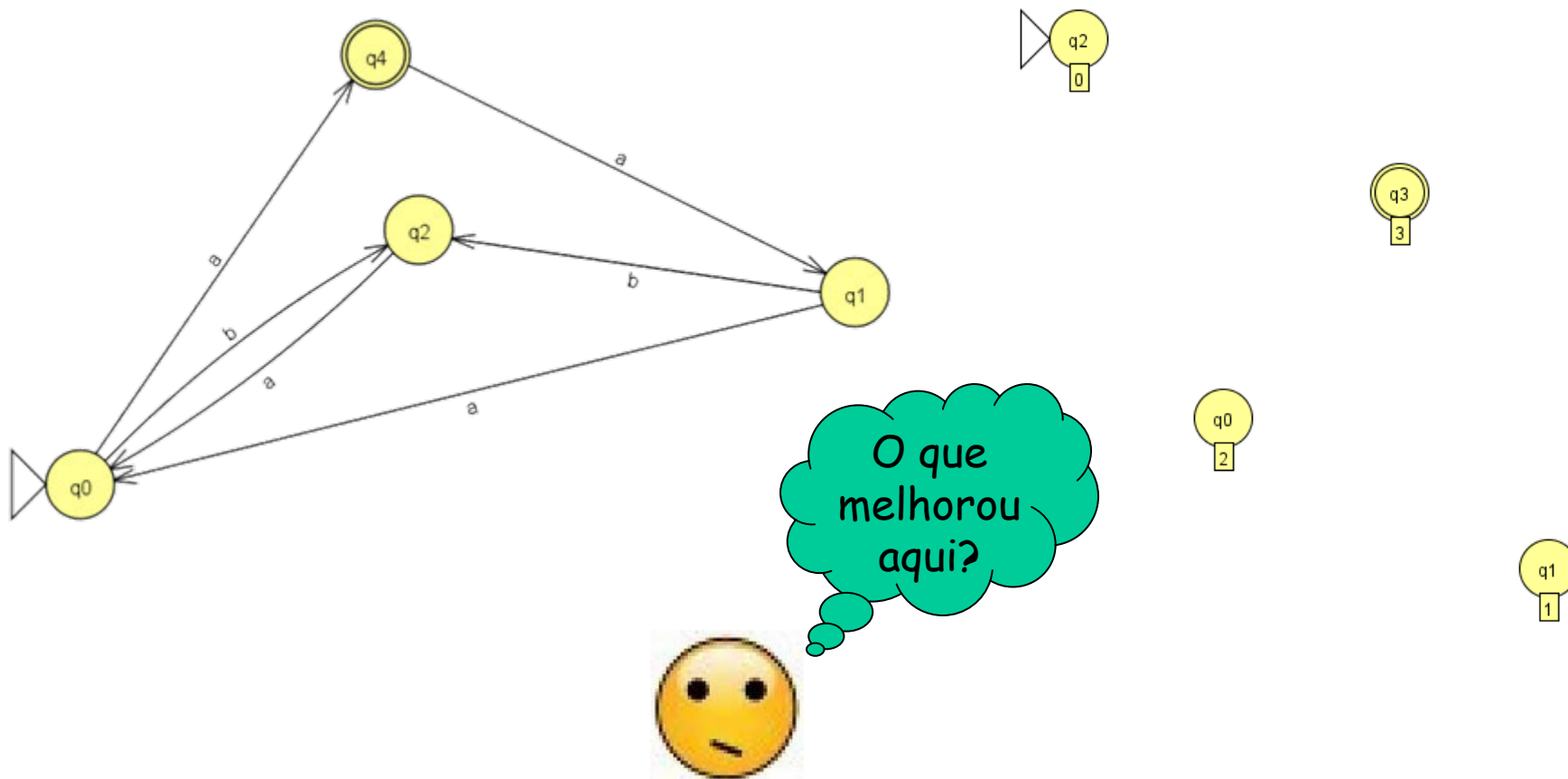
Qual a próxima partição?



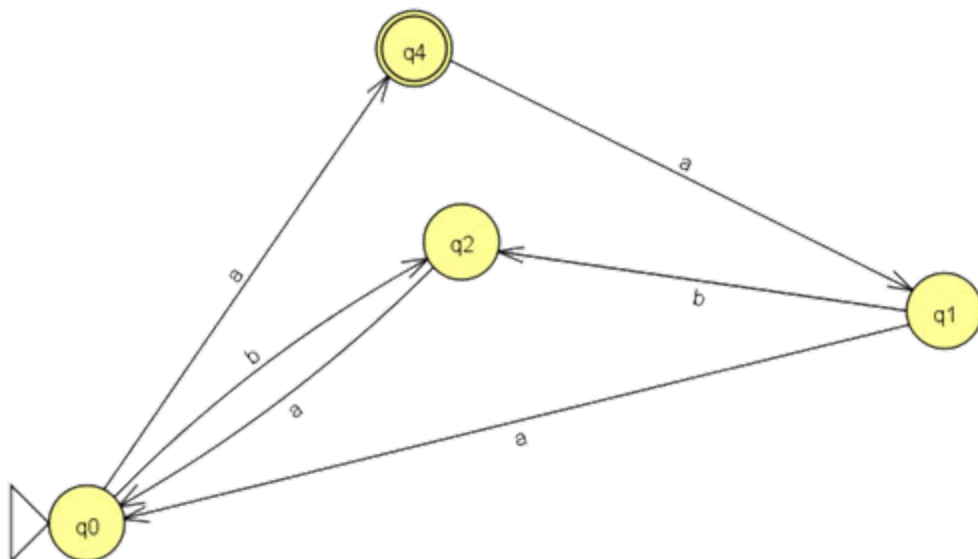
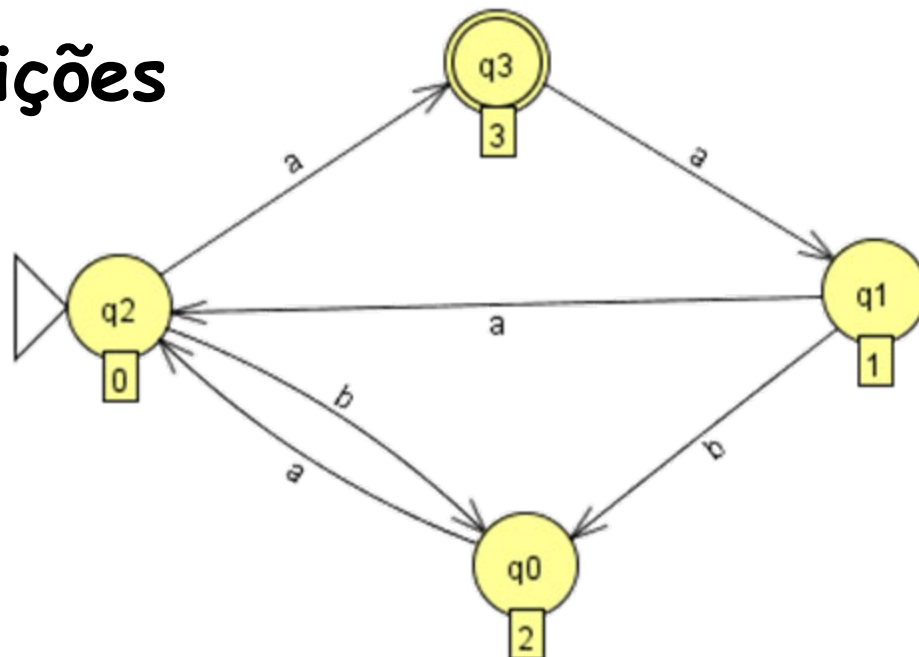
## Nós folha completamente definidos



## Construção das Transições

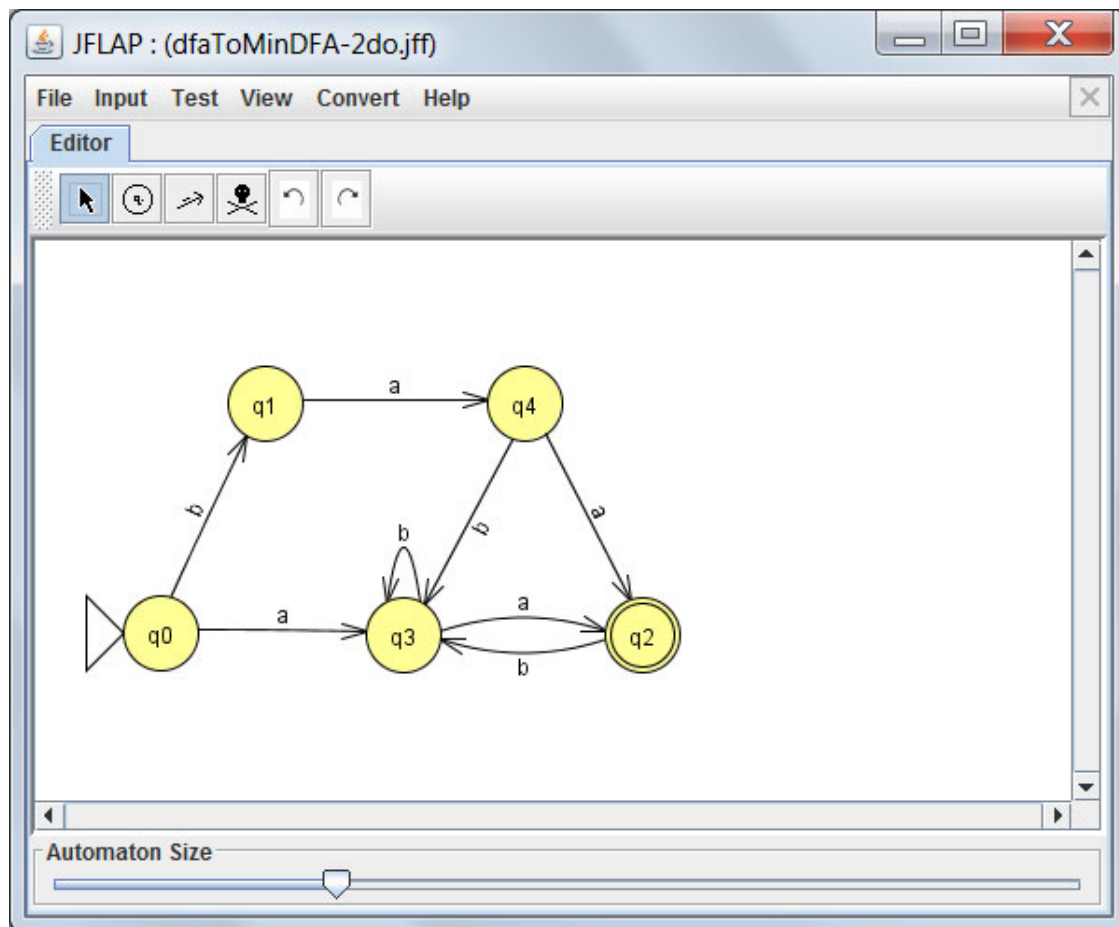


## Construção das Transições



O autômato já  
era o mínimo.  
Quando deu para  
ver?

## Agora um autômato não mínimo para valer



Defina a linguagem

(adicione o trap state)

Construa as partições

Tome o conjunto mínimo de estados (sem o trap state, que é inútil por definição)

Complete as transições necessárias

## Novo caso a minimizar (Tutorial do JFLAP)

Este autômato está longe do mínimo

