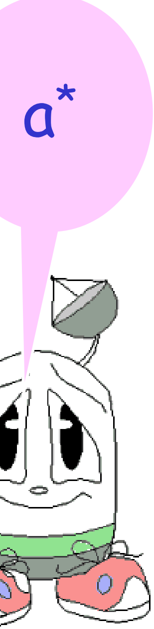


Automatos Finitos

Algoritmo de Minimização de AFD

(Menezes, 2002) e no Livro Animado do próprio autor:

ttp://teia.inf.ufrgs.br/cgi-bin/moore.pl?curso=LivroAnimado&estado=81

Autômato Finito Mínimo

- Um Autômato Mínimo é útil em muitas soluções práticas.
- Entretanto, em algumas aplicações especiais, a minimização do número de estados não implica necessariamente no menor custo de implementação.
- Um exemplo típico seria o desenho de circuitos eletrônicos, quando pode ser desejável introduzir estados intermediários em determinadas posições (do circuito), de forma a melhorar a eficiência, ou simplesmente facilitar as ligações físicas.
- Portanto, o algoritmo de minimização nestes casos deve ser modificado, prevendo as variáveis específicas da aplicação.

Como o algoritmo funciona

- O algoritmo de minimização unifica os estados equivalentes de um autômato.

- Dois estados q e p são ditos equivalentes se, e somente se, para qualquer cadeia w pertencente a Σ^*, $\delta(q,w)$ e $\delta(p, w)$ resultam simultaneamente em estados finais, ou não-finais.

- Ou seja, o processamento de uma entrada qualquer a partir de estados equivalentes gera, em qualquer caso, o mesmo resultado: aceitação ou rejeição.

- Se dois estados não são equivalentes, eles são ditos distingüíveis.

Minimização de um AF

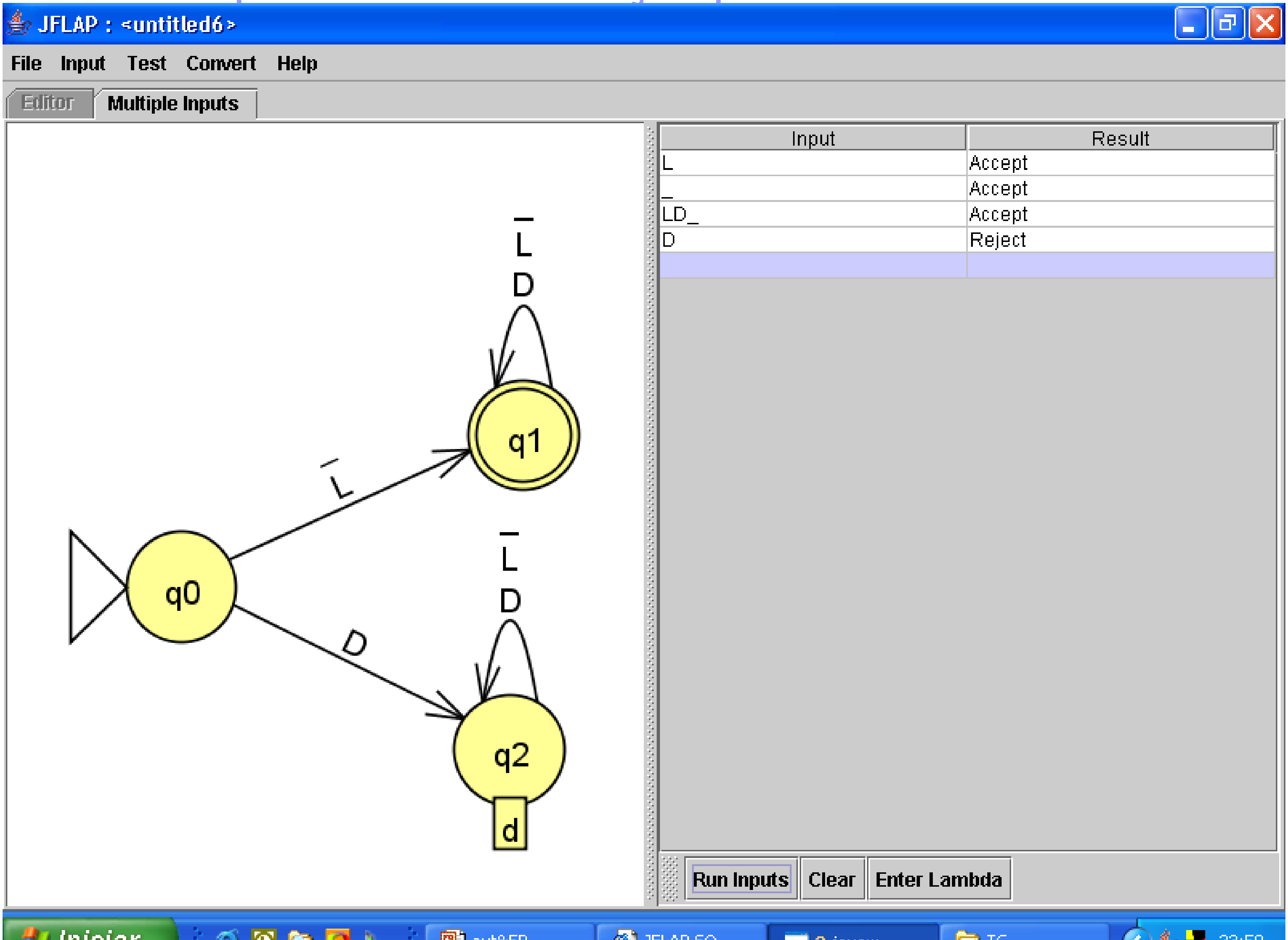
- ***Def:*** Um autômato mínimo de uma LR é um AFD com um número X de estados tal que qualquer outro AFD que aceita a mesma linguagem terá um número de estados maior ou igual a X.

- Pré-requisitos para um AF ser minimizado:

a) Ser determinístico
b) Não pode ter estados inacessíveis
c) δ deve ser total (cada estado deve possuir transições para todos os elementos do alfabeto de entrada). Deve ser um AFD no senso estrito.

- Caso o AF não possua algum dos requisitos acima é necessário gerar um AF equivalente.
- No caso do item c) devemos incluir um novo estado d para incluir as transições não previstas e incluir um ciclo em d para todos os símbolos de Σ.

Exemplo de transformação para um AFD estrito



Algoritmo de minimização

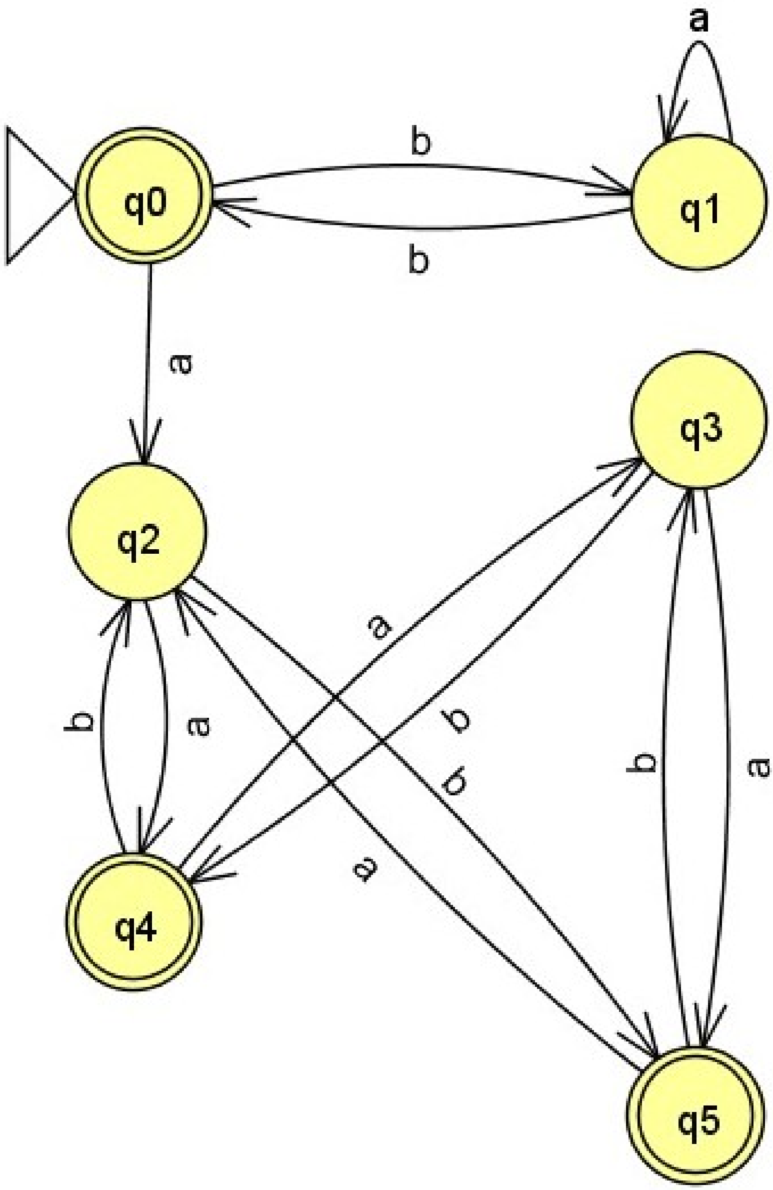
Identifica os estados equivalentes por exclusão.

A partir de uma tabela de estados, são marcados os não-equivalentes.

Ao final do algoritmo, os itens não-marcados representam os estados equivalentes.

q1					
q2					
...					
qm					
d					
	q0	q1		qm-1	qm

Autômato a ser minimizado



1) Construir a tabela com cada par de estados ocorrendo 1 vez

q1					
q2					
q3					
q4					
q5					
	q0	q1	q2	q3	q4

2) Marcar estados trivialmente não-equivalentes {estado final, estado não-final}

q1	X				
q2	X				
q3	X				
q4		X	X	X	
q5		X	X	X	
	q0	q1	q2	q3	q4

3) Marcar estados não-equivalentes

- Para cada par {qu,qv} não-marcado e para cada símbolo a $\in\Sigma$, suponha que δ(qu,a) = pu e δ(qv,a) = pv e:

- Se pu = pv, então qu é equivalente a qv para o símbolo a e não deve ser marcado
- Se pu <> pv e o par {pu, pv} não está marcado, então {qu, qv} é incluído em uma lista a partir de {pu, pv} para análise posterior
- Se pu <> pv e o par {pu, pv} está marcado, então marcar todos os pares da lista (e, recursivamente se algum par da lista encabeça outra lista)

Usamos (+) para marcar os pares desta etapa

{q0,q4}: $\delta(q0,a) = q2$ $\delta(q0,b) = q1$

$\delta(q4,a) = q3$ $\delta(q4,b) = q2$

omo {q1,q2} e {q2,q3} são não-
arcados, então {q0,q4} é incluído
as listas encabeçadas por {q1,q2}
{q2,q3}

q1	X				
q2	X				
q3	X				
q4		X	X	X	
q5		X	X	X	
	q0	q1	q2	q3	q4

2) {q0,q5}: $\delta(q0,a) = q2$ $\delta(q0,b) = q1$

$\delta(q5,a) = q2$ $\delta(q5,b) = q3$

Como {q1,q3} é não-marcado (e {q2,q2} é triv. eq.) então {q0,q5} é incluído na lista encabeçada por {q1,q3}

{q0,q5}

{q0,q4}

{q0,q4}

3) {q1,q2}: $\delta(q1,a) = q1$ $\delta(q1,b) = q0$

$\delta(q2,a) = q4$ $\delta(q2,b) = q5$

Como {q1,q4} é marcado então {q1,q2} também deve ser marcado. Como {q1,q2} encabeça uma lista, o par {q0,q4} também é marcado

Usamos (+) para marcar os pares desta etapa

) {q0,q4}: $\delta(q0,a) = q2$ $\delta(q0,b) = q1$

$\delta(q4,a) = q3$ $\delta(q4,b) = q2$

omo {q1,q2} e {q2,q3} são não-
arcados, então {q0,q4} é incluído
as listas encabeçadas por {q1,q2}
{q2,q3}

2) {q0,q5}: $\delta(q0,a) = q2$ $\delta(q0,b) = q1$

$\delta(q5,a) = q2$ $\delta(q5,b) = q3$

Como {q1,q3} é não-marcado (e {q2,q2} é triv. eq.) então {q0,q5} é incluído na lista encabeçada por {q1,q3}

q1	X				
q2	X	(+)			
q3	X				
q4	(+)	X	X	X	
q5		X	X	X	
	q0	q1	q2	q3	q4

{q0,q5}

{q0,q4}

{q0,q4}

3) {q1,q2}: $\delta(q1,a) = q1$ $\delta(q1,b) = q0$

$\delta(q2,a) = q4$ $\delta(q2,b) = q5$

Como {q1,q4} é marcado então {q1,q2} também é marcado. Como {q1,q2} encabeça uma lista, o par {q0,q4} também é marcado

) {q1,q3}: $\delta(q1,a) = q1$ $\delta(q1,b) = q0$

$\delta(q3,a) = q5$ $\delta(q3,b) = q4$

omo {q1,q5} e {q0,q4} são
arcados, então {q1,q3} também é
arcado. Como {q1,q3} encabeça uma
sta, {q0,q5} também é maracdo

q1	X				
q2	X	(+)			
q3	X				
q4	(+)	X	X	X	
q5		X	X	X	
	q0	q1	q2	q3	q4

{q0,q5}

{q0,q4}

{q0,q4}

) {q1,q3}: $\delta(q1,a) = q1$ $\delta(q1,b) = q0$

$\delta(q3,a) = q5$ $\delta(q3,b) = q4$

omo {q1,q5} e {q0,q4} são
arcados, então {q1,q3} também é
arcado. Como {q1,q3} encabeça uma
sta, {q0,q5} também é marcado

5) {q2,q3}: $\delta(q2,a) = q4$ $\delta(q2,b) = q5$

$\delta(q3,a) = q5$ $\delta(q3,b) = q4$

Como {q4,q5} é não-marcado então
{q2,q3} é incluído na lista encabeçad
por {q4,q5}

q1	X				
q2	X	(+)			
q3	X	(+)			
q4	(+)	X	X	X	
q5	(+)	X	X	X	
	q0	q1	q2	q3	q4

{q0,q5}

{q0,q4}

{q0,q4}

{q2,q3}

) {q1,q3}: $\delta(q1,a) = q1$ $\delta(q1,b) = q0$

$\delta(q3,a) = q5$ $\delta(q3,b) = q4$

omo {q1,q5} e {q0,q4} são
arcados, então {q1,q3} também é
arcado. Como {q1,q3} encabeça uma
sta, {q0,q5} também é marcado

q1	X				
q2	X	(+)			
q3	X	(+)			
q4	(+)	X	X	X	
q5	(+)	X	X	X	
	q0	q1	q2	q3	q4

5) {q2,q3}: $\delta(q2,a) = q4$ $\delta(q2,b) = q5$

$\delta(q3,a) = q5$ $\delta(q3,b) = q4$

Como {q4,q5} é não-marcado então
{q2,q3} é incluído na lista encabeçad
por {q4,q5}

{q0,q5}

{q0,q4}

{q0,q4} {q4,q5}

6) {q4,q5}: $\delta(q4,a) = q3$ $\delta(q4,b) = q2$

{q2,q3} $\delta(q5,a) = q2$ $\delta(q5,b) = q3$

Como {q2,q3} é não-marcado então
{q4,q5} é incluído na lista
encabeçada por {q2,q3}

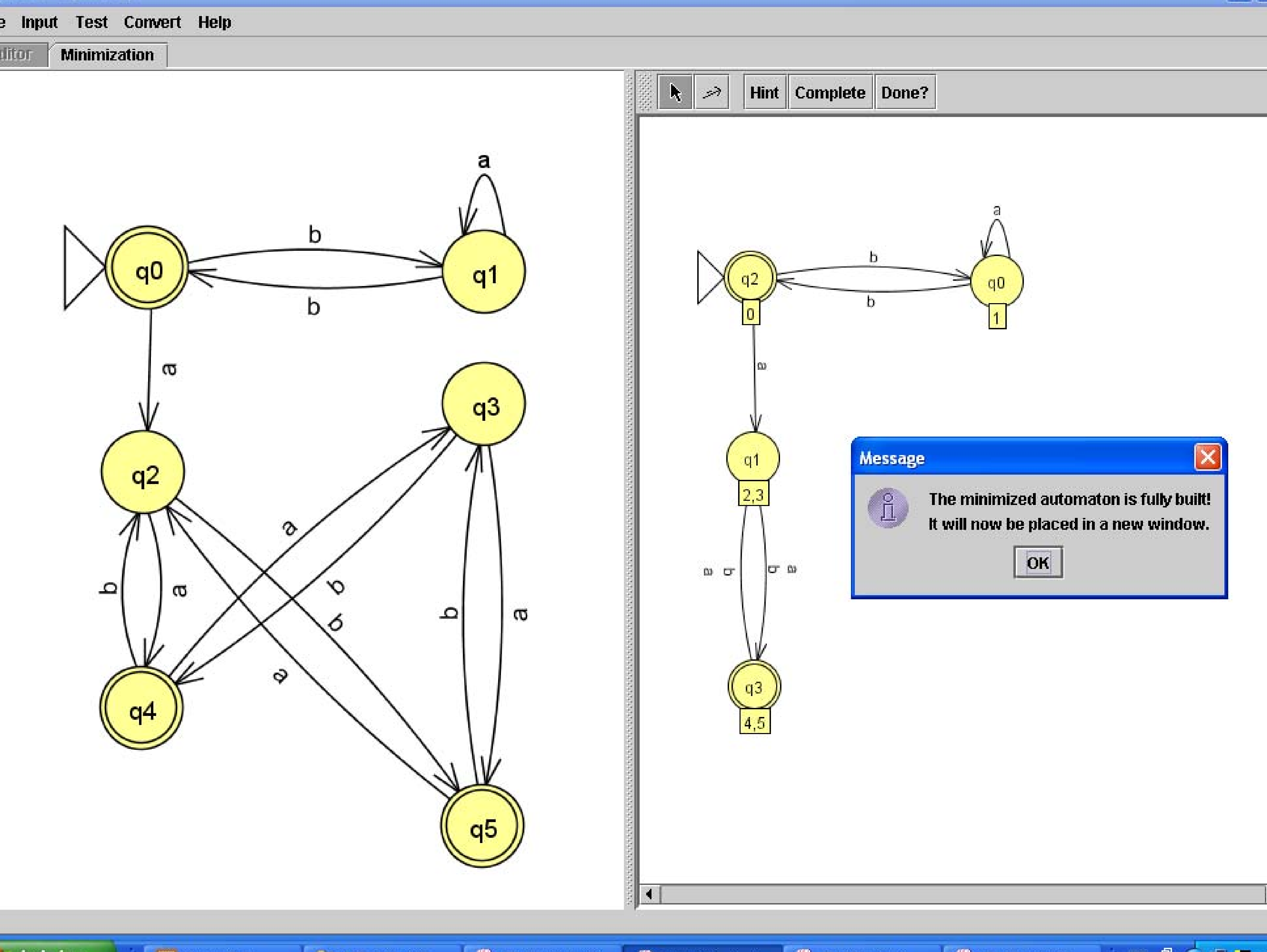
Unificações

- Como os pares {q2,q3} e {q4,q5} são não marcados, as seguintes unificações podem ser feitas:

  - q23 representa a unificação dos estados não-finais q2 e q3
  - q45 representa a unificação dos estados finais q4 e q5

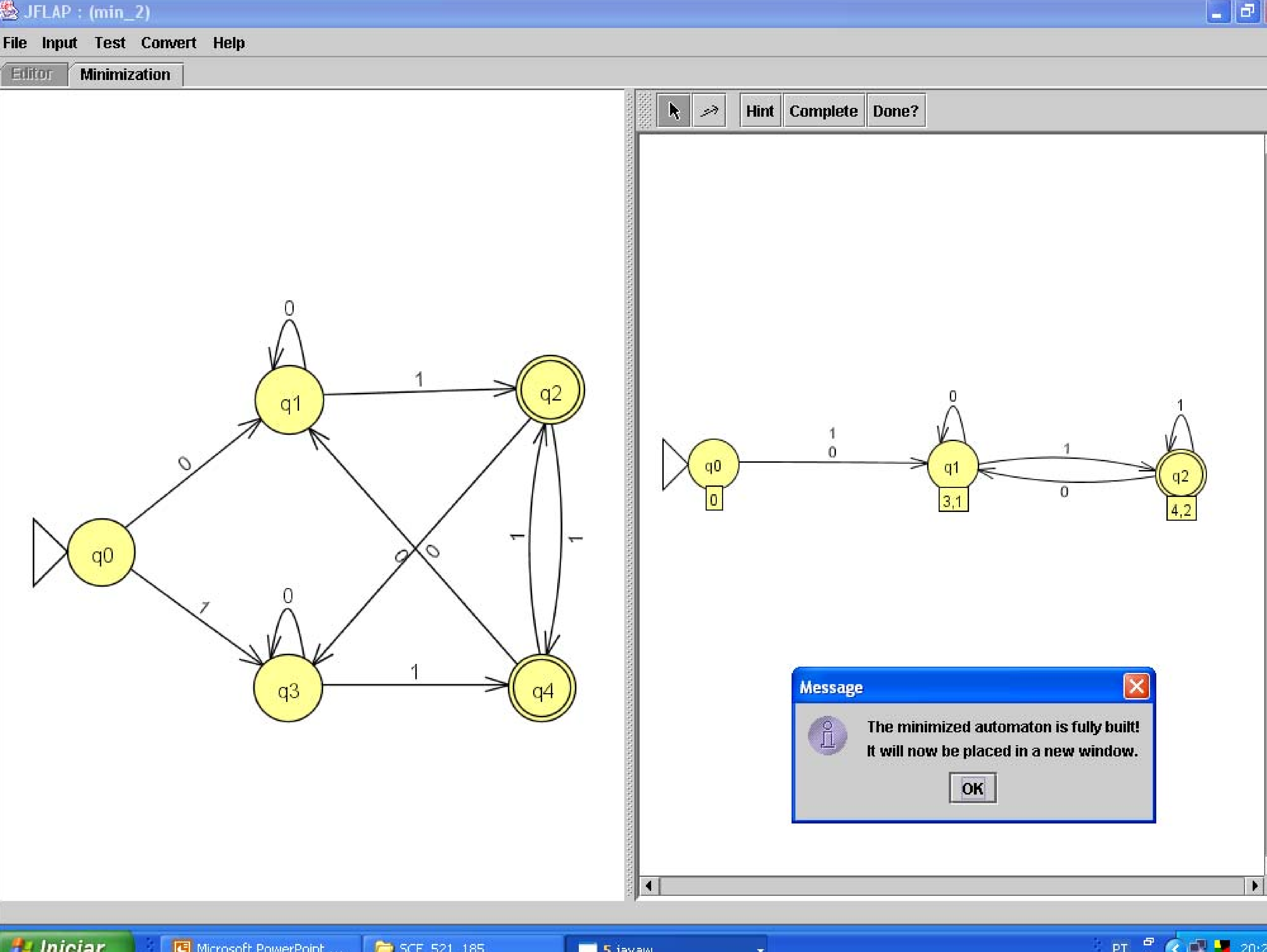
- Tempo do algoritmo de minimização= tempo para preencher as entradas para os $m(m-1)/2$ pares de estados (combinações 2 a 2). Ou seja, $O(m^2)$

- Teo 2.28 (Menezes, 2002) O AFD mínimo de uma linguagem é único.


Input
Test
Convert
Help
Minimization
Hint
Complete
Done?
q0
q1
q2
q3
q4
q5
a
b
0
1
2,3
4,5
Message
The minimized automaton is fully built!
It will now be placed in a new window.
OK

Exercício para casa

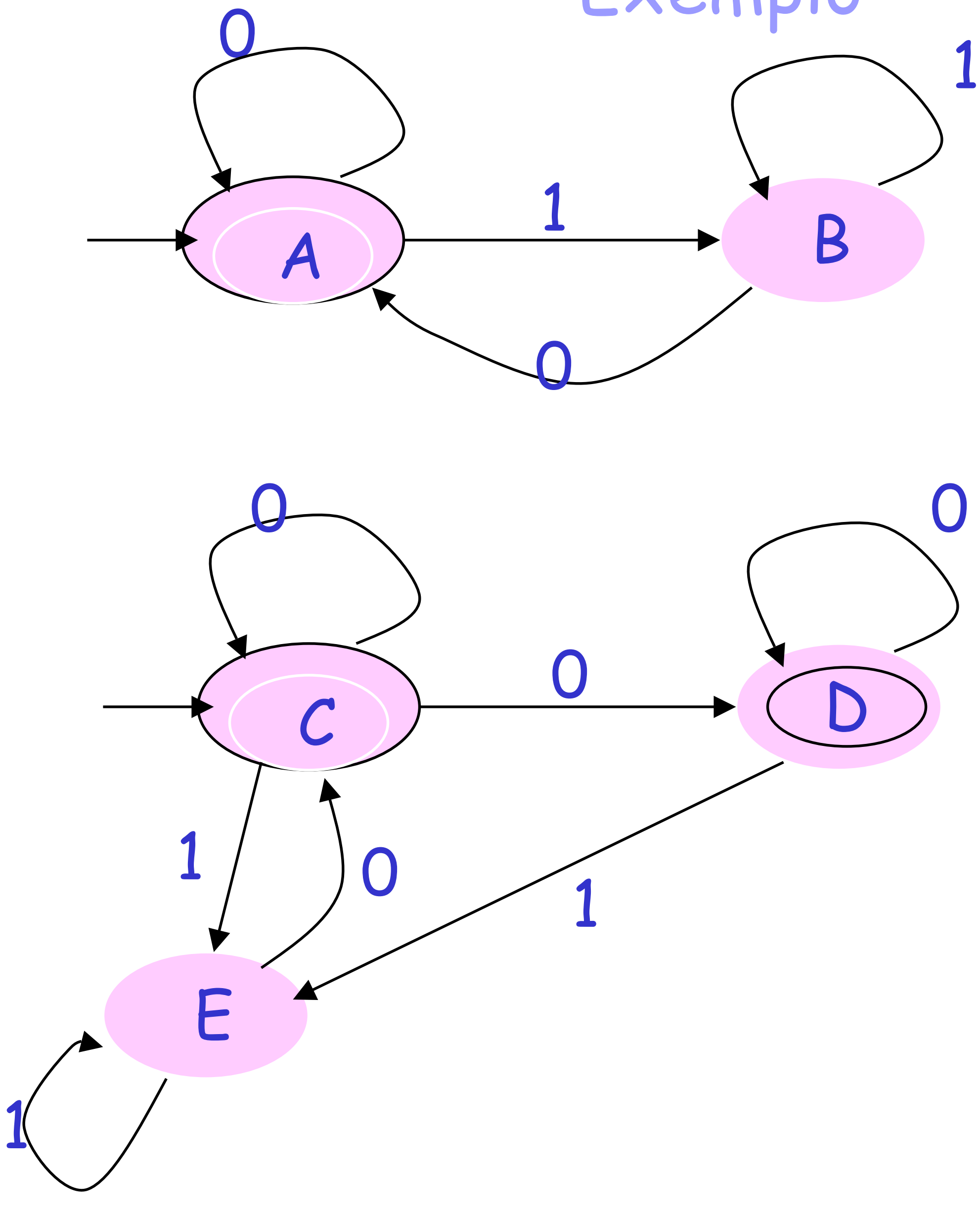
- Apliquem o algoritmo para o AF a seguir e compare com o resultado do JFLAP mostrado ao lado.


JFLAP : (min_2)
File Input Test Convert Help
Editor
Minimization
Hint
Complete
Done?
q0
q1
q2
q3
q4
0
1
0
3,1
4,2
Message
The minimized automaton is fully built!
It will now be placed in a new window.
OK
Iniciar
Microsoft PowerPoint ...
SCE 521 185
5 javaw
PT

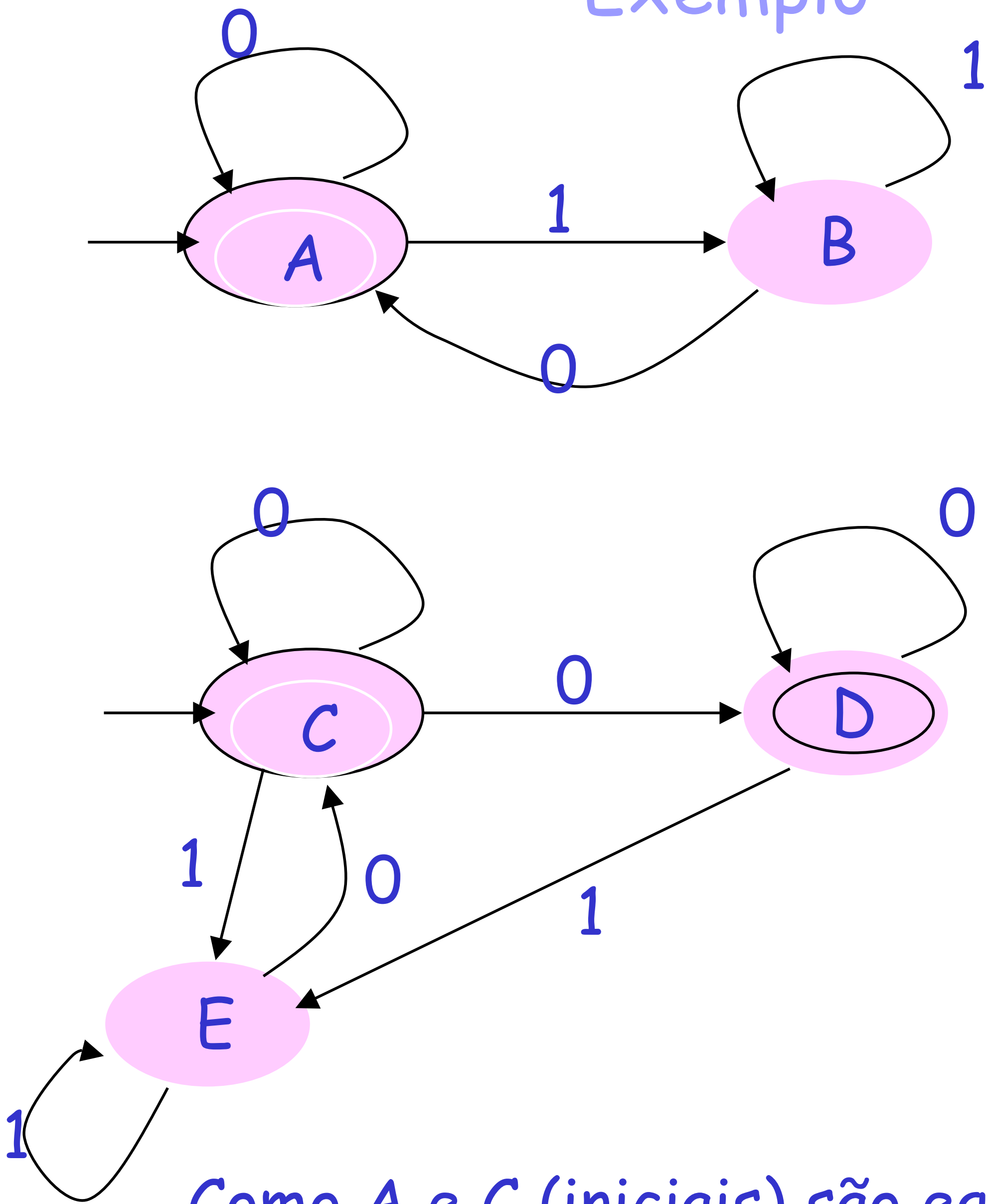
Verificando Equivalência de Linguagens Regulares

- Converta as representações em AFDs.
- Imagine um AFD que é a união dos estados de todos os AFDs iniciais.
- Preencha a tabela de equivalência para esse autômato.
- Teste se os estados iniciais dos AFD originais são equivalentes. Se sim, então as linguagens são iguais; c.c. são diferentes.

Exemplo



Exemplo



	A	B	C	D
B	X			
C		X		
D		X		
E	X		X	X

Como A e C (iniciais) são equivalentes, então os AFDs aceitam a mesma linguagem

Resumo Parcial

- *O lema do bombeamento:* se uma linguagem é regular, então toda cadeia suficientemente longa tem uma subcadeia não-vazia que pode ser "bombeada", i.e, repetida qualquer número de vezes enquanto as cadeias resultantes também estão na linguagem. Este fato pode ser usado para provar que muitas linguagens diferentes não são regulares.
- *Operações que preservam a propriedade de ser uma linguagem regular*: união, concatenação, fechamento, interseção, complementação, diferença, reversão.
- *Como testar o caráter vazio de linguagens regulares:* há um algoritmo que, dada uma representação de uma LR, como um autômato ou uma ER, informa se a linguagem é ou não o conjunto vazio.
- *Como testar a pertinência em uma linguagem regular:* há um algoritmo que, dada uma cadeia e uma representação de uma LR, informa se a cadeia pertence ou não à linguagem.

- *Como testar a distinção de estados:* Dois estados de um AFD são distinguíveis se existe uma cadeia que leva exatamente um dos dois estados para um estado de aceitação. Começando apenas com o fato de que pares que consistem em um estado de aceitação e um de não-aceitação são distinguíveis, e tentando descobrir pares de estados distinguíveis adicionais encontrando pares cujos sucessores sobre um símbolo de entrada são distinguíveis, podemos descobrir todos os pares de estados distinguíveis.
- *Minimização de AFDs*: Podemos particionar os estados de qualquer AFD em grupos de estados mutuamente indistinguíveis. Elementos de dois grupos diferentes são sempre distinguíveis. Se substituirmos cada grupo por um único estado, obteremos um AFD equivalente que tem tão poucos estados quanto qualquer AFD para a mesma linguagem.