



# Expressões Regulares

## Expressões Regulares (ER)

Conversão de AF para ER no JFLAP

Equivalências entre AFD, AFND, AF- $\lambda$ , ER, GR

# Expressões Regulares (ER)

Uma ER sobre um alfabeto  $\Sigma$  é definida como:

- a)  $\emptyset$  é uma ER e denota a linguagem vazia
- b)  $\lambda$  é uma ER e denota a linguagem contendo a palavra vazia, ie  $\{\lambda\}$
- c) Qualquer símbolo  $x \in \Sigma$  é uma ER e denota a linguagem  $\{x\}$
- d) Se  $r$  e  $s$  são ER denotando as linguagens  $R$  e  $S$  então:
  - $(r+s)$  ou  $(r|s)$  é ER e denota a linguagem  $R \cup S$
  - $(rs)$  é ER e denota a linguagem  $RS = \{w \mid u \in R \text{ e } v \in S\}$
  - $(r^*)$  é ER e denota a linguagem  $R^*$

# Exemplos

1.  $00$  é uma ER denotando a linguagem  $\{00\}$
2.  $(0+1)^*$  denota a linguagem formada por todas as cadeias de 0's e 1's =  $\{\lambda, 0, 1, 00, 01, 10, 11, 000, \dots\}$  Pode ser também  $(a^*b^*)^*$
3.  $(0+1)^* 00 (0+1)^*$  denota todas as cadeias de 0's e 1's com ao menos dois 0's consecutivos
4.  $a+b^*c$  denota um único  $a$  e todas as cadeias consistindo de zero ou mais vezes  $b$  seguido de  $c$ . A linguagem é formada por  $\{a, c, bc, bbc, bbbc, \dots\}$

# Exemplo 4 - JFLAP

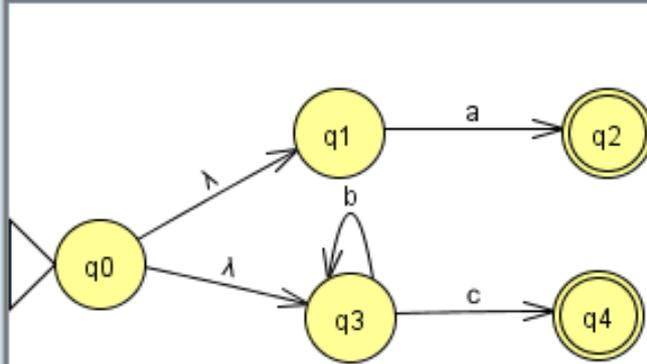
JFLAP : <untitled6>

File Input Test Convert Help

Editor Convert FA to RE

Make Single Noninitial Final State  
Create a new state to make a single final state.

Do It Export



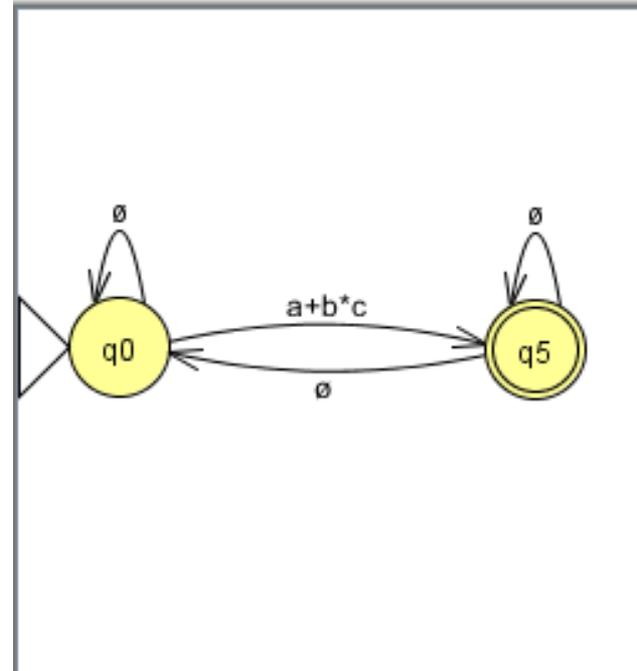
JFLAP : <untitled6>

File Input Test Convert Help

Editor Convert FA to RE

Generalized Transition Graph Finished!  
 $a+b^*c$

Do It Export



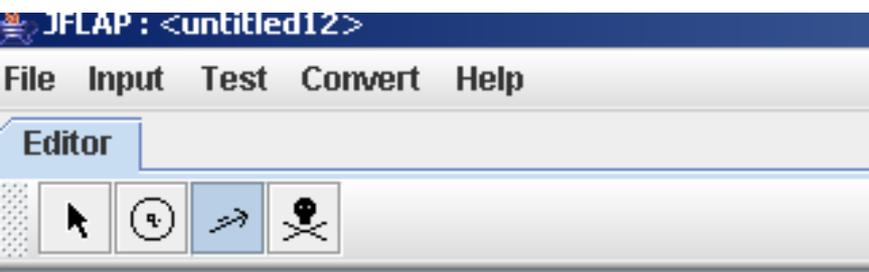
JFLAP : <untitled7>

File Convert Help

Editor

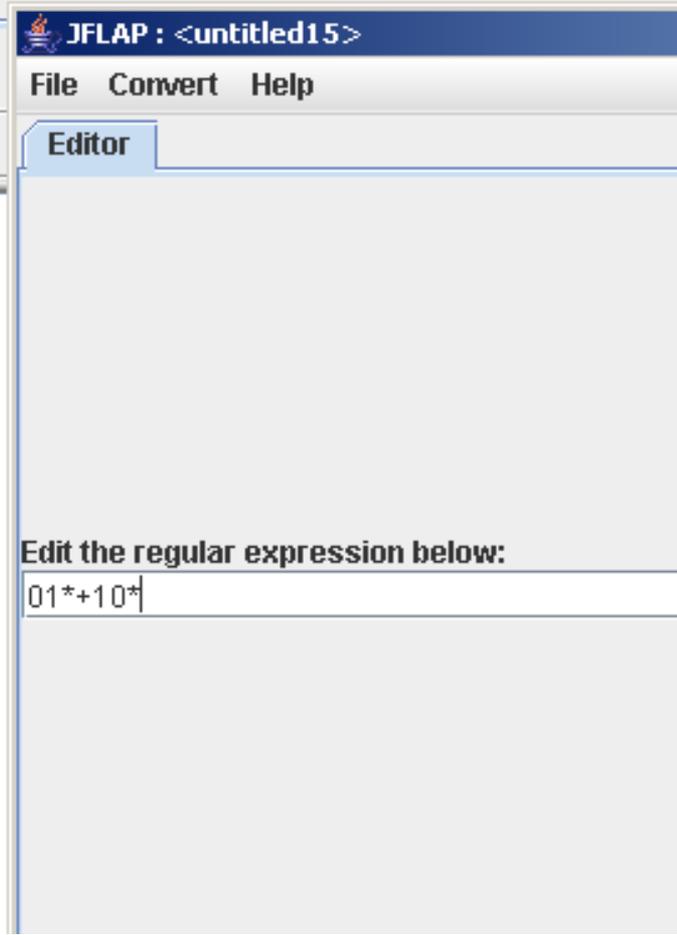
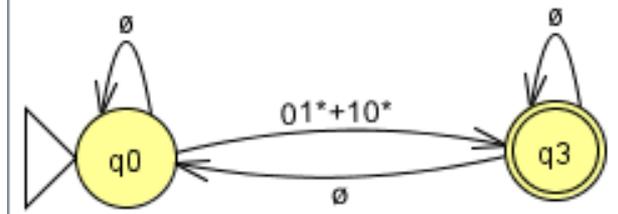
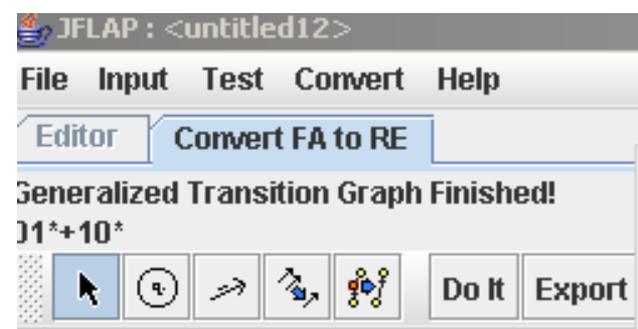
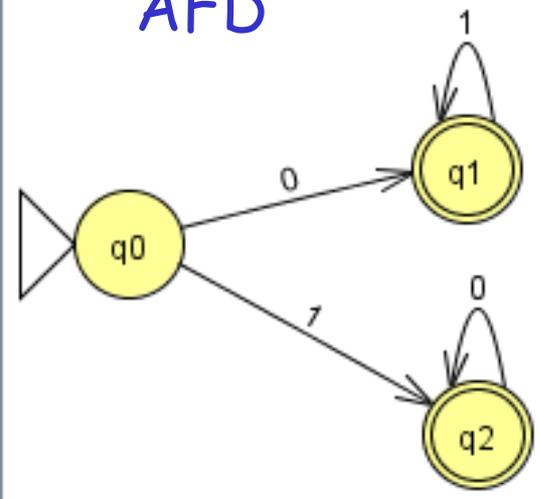
Edit the regular expression below:  
 $a+b^*c$

- $(0+1)^* 001$  denota todas as cadeias de 0's e 1's terminadas em 001
- $0^*1^*2^*$  denota qualquer número de 0's seguido por qualquer número de 1's seguido por qualquer número de 2's
- $01^* + 10^*$  denota a linguagem consistindo de todas as cadeias que são um único 0 seguido por qualquer número de 1's e um único 1 seguido por qualquer número de 0's =  $\{0,01,011,\dots,1,10,100,\dots\}$

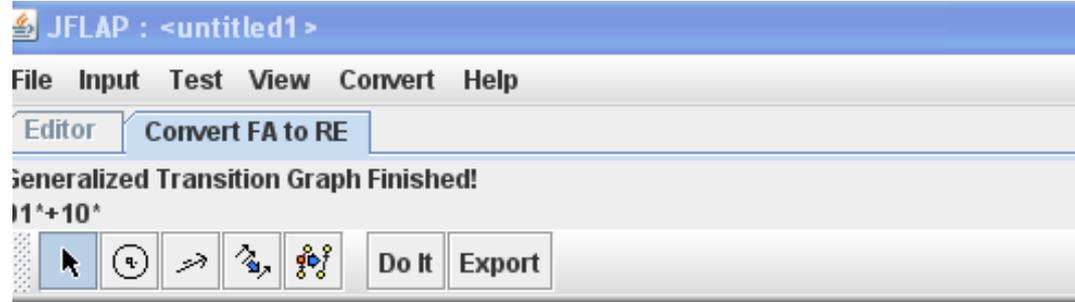
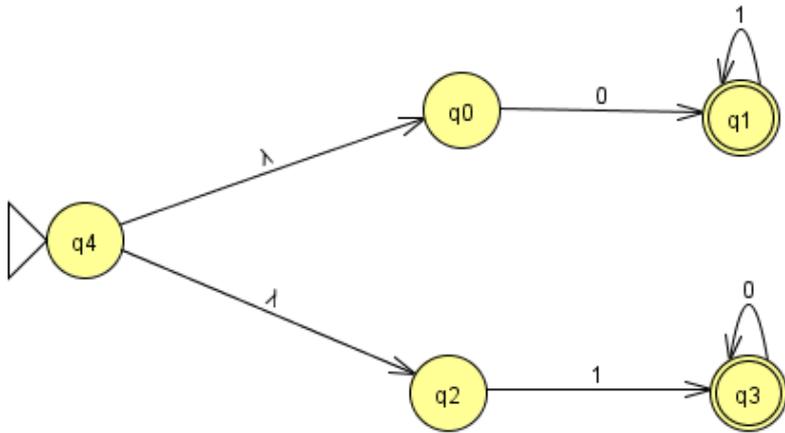


$$01^* + 10^*$$

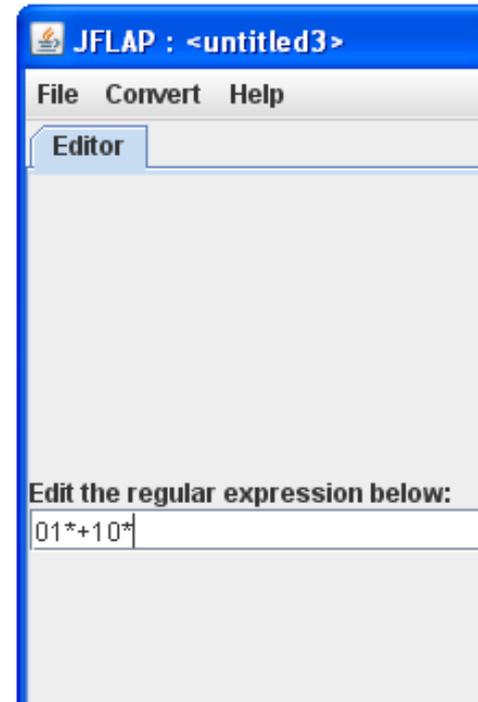
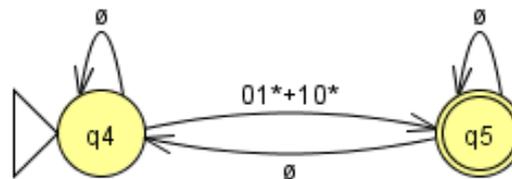
AFD



$01^* + 10^*$



AF- $\lambda$



## Omissão de parênteses

- Para omitir parênteses devemos respeitar:
  - O fecho (\*) tem prioridade sobre a concatenação (rs) que tem prioridade sobre a união.
  - A concatenação e a união são associadas da esquerda para a direita.
  - Ex:  $01^* + 1$  é agrupado como  $(0(1^*)) + 1 \Rightarrow L = \{1, 0, 01, 011, \dots\}$
- Usamos parênteses quando queremos alterar a prioridade:
- $(01)^* + 1 \Rightarrow L = \{1 \cup (01)^n \mid n \geq 0\} = \{1, \lambda, 01, 0101, \dots\}$
- $0(1^* + 1) \Rightarrow L = \{w \in \{0,1\}^* \mid w \text{ começa com } 0 \text{ seguido de } 1^n \mid n \geq 0\} \rightarrow \text{Lei distributiva à esq} = 01^* + 01 = \{0, 01, 011, 0111, \dots\}$

## Exercícios

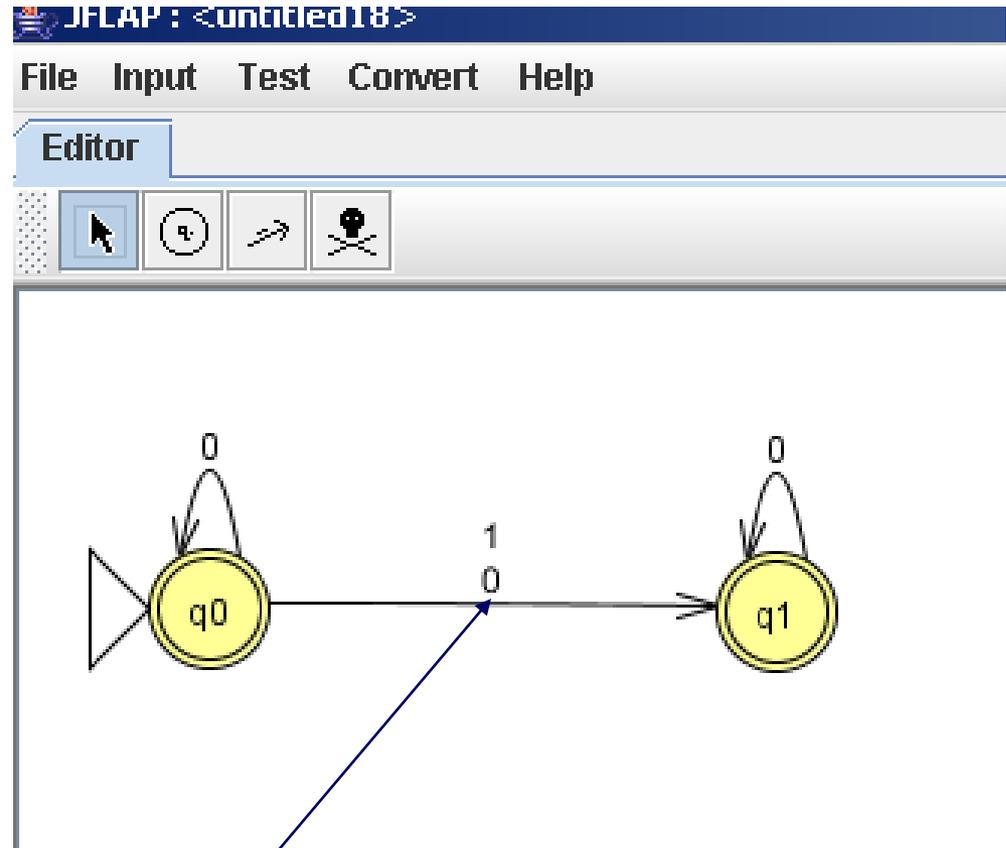
1. O conjunto de cadeias sobre  $\{0,1\}$  que termine com três 1's consecutivos.
2. O conjunto de cadeias sobre  $\{0,1\}$  que tenha pelo menos um 1.
3. O conjunto de cadeias sobre  $\{0,1\}$  que tenha no máximo um 1.

## Exemplo 3

- O conjunto de cadeias sobre  $\{0,1\}$  que tenha no máximo um 1.

$0^*(\lambda+(0+1)0^*) \rightarrow$   
podem usar  $\lambda$  para  
simplificar a ER que  
seria:

$0^* (1+0) 0^* + 0^*$



O não é  
necessário

JFLAP : <untitled18>

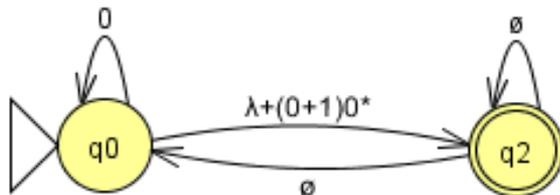
File Input Test Convert Help

Editor Convert FA to RE

Generalized Transition Graph Finished!

$0^*(\lambda+(0+1)0^*)$

⏪ ⏩ ↻ ↶ ↷ ⚙ Do It Export



JFLAP : <untitled20>

File Convert Help

Editor

Edit the regular expression below:

$0^*(\lambda+(0+1)0^*)$

# Conversão de AF para ER no JFLAP

File Input Test Convert Help

Editor Convert FSA to RE

Make Single Noninitial Final State  
Create a new state to make a single final state.

Do It Export

Insira um novo estado final como pede o JFLAP

$L(M1) = \{x \in \{1\}^* \mid x \text{ possui um nro par de } 1\text{'s}\}.$

Windows taskbar: Iniciar, Journal of Education..., http://teia.inf.ufrgs..., Admin, aut\_min.ppt, aut3.ppt, JFLAP : <untitled3>, JFLAP : <untitled5>, 09:09

JFLAP : <untitled3>

File Input Test Convert Help

Editor Convert FSA to RE

Make Single Noninitial Final State  
Put lambda transitions from old final states to new.

Do It Export

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

Clique em  
Do it!

# Coloque transições nulas nos estados sem transição

JFLAP: <untitled3>

File Input Test Convert Help

Editor Convert FSA to RE

Reform Transitions  
Put empty transitions between states with no transitions. 6 more empty transitions needed.

Do It Export

q0 q1 q2

1 1  $\lambda$

Clique em  
Do it!

Iniciar Journal of Educational T... http://teia.inf.ufrgs.br/c... aula8 JFLAP: <untitled3> Microsoft PowerPoint - [...]

09:46

# Completa as 6 transições faltantes

JFLAP : <untitled3>

File Input Test Convert Help

Editor Convert FSA to RE

Remove States  
Use the collapse state tool to remove nonfinal, noninitial states. 1 more removals needed.

Do it Export

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

Clique em  
Do it!

Iniciar Journal of Educational T... http://teia.inf.ufrgs.br/c... aula8 JFLAP : <untitled3> Microsoft PowerPoint - [...]

09:48

# Grafo terminado

The image shows a screenshot of the JFLAP software interface. The main window, titled "JFLAP : <untitled3>", displays a transition graph with two states,  $q_0$  and  $q_2$ . State  $q_0$  is the start state, indicated by a double circle. It has a self-loop labeled "11" and a transition to state  $q_2$  labeled "1". State  $q_2$  is an accepting state, indicated by a double circle, and has a self-loop labeled "0".

Below the graph, a smaller window titled "JFLAP : <untitled6>" is shown. It has a menu bar with "File", "Convert", and "Help". The "Editor" tab is active, and it contains the text "Edit the regular expression below:" followed by a text input field containing the regular expression  $(11)^*$ . A large black arrow points from the text input field down to the regular expression  $(11)^*$  displayed in blue text below the main window.

The Windows taskbar at the bottom shows several open applications: "Iniciar", "Journal of Educational T...", "http://teia.inf.ufrgs.br/c...", "aula8", "JFLAP : <untitled3>", "JFLAP : <untitled6>", "Microsoft PowerPoint - [...]", and the system clock showing "09:51".

## Exercícios

- O conjunto de cadeias de 0's e 1's com pelo menos um par de 1's consecutivos
- O conjunto de cadeias de 0's e 1's com no máximo um par de 1's consecutivos
- O conjunto de 0's e 1's cujo 5º símbolo é 1
- O conjunto de cadeias sobre  $\{a,b,c\}$  contendo pelo menos um a e pelo menos um b

# Solução

- O conjunto de cadeias de 0's e 1's com pelo menos um par de 1's consecutivos:  $(0+1)^*11(1+0)^*$
- O conjunto de cadeias de 0's e 1's com no máximo um par de 1's consecutivos:  $(0+1)^*(\lambda+11(1+0)^*)$  não funciona ...pois o fecho dá origem a mais de um 11. Analisem a próxima solução:  
$$\lambda + 11 + ((10)^* 11 (01)^*) + (0^* 11 0^*)$$
- O conjunto de 0's e 1's cujo 5º símbolo é 1:  $(0+1)(0+1)(0+1)(0+1)1(0+1)^*$
- O conjunto de cadeias sobre  $\{a,b,c\}$  contendo pelo menos um a e pelo menos um b. Analisem as 2 soluções e vejam se são iguais:

$$((a+b+c)^*ab(a+b+c)^*) + ((a+b+c)^*ba(a+b+c)^*)$$

$$c^*a(a+c)^*b(a+b+c)^* + c^*b(b+c)^*a(a+b+c)^*$$

## Propriedades algébricas das ER

- $L + M = M + L$  (**união** é comutativa)
  - $(L + M) + N = L + (M + N)$  (**união** é associativa)
  - $(LM)N = L(MN)$  (**concatenação** é associativa)
  - Exercício: a **concatenação** é comutativa???
- 
- $\emptyset + L = L + \emptyset = L$  ( $\emptyset$  é o elemento nulo para união)
  - $\lambda L = L\lambda = L$  ( $\lambda$  é o elemento nulo para concatenação)
  - $\emptyset L = L\emptyset = \emptyset$

- $L(M + N) = LM + LN$  (lei distributiva à esq)
- $(M + N)L = ML + NL$  (lei distributiva à dir)
- $L + L = L$
- $(L^*)^* = L^*$
- $\emptyset^* = \lambda$
- $\lambda^* = \lambda$
- Algumas extensões de LR usadas em utilitários do UNIX
- $L^+ = LL^*$
- $L? = (L + \lambda)$  (usado no Lex para indicar opcional)
- $L^* = L^+ + \lambda$

# Exercícios

- Faça ER para:
  - identificadores
  - números reais com sinais
  - Inteiros com sinais
  - cadeias de caracteres
  - e comentários do **Pascal**.
- 
- reais do **Fortran** (.5 e 5. além dos padrões de reais de Pascal)
- 
- *OBS: Geralmente, nas modelagens de Linguagem de Programação não usamos sinais junto com o valor das constantes inteiras.*

- Pascal, com  $L = \{a..z, A..Z\}$ ;  $D = \{0..9\}$
  - ID:  $(L|_)(L|D|_)^*$
  - Reais:  $(+|-|\lambda) (D^+ . D^+ (E (+|-|\lambda) D^+ | \lambda) | D^+ (. D^+ | \lambda) E (+|-|\lambda) D^+ )$
  - Observem que acima exigimos que o real tenha uma parte com ponto fixo **ou** com ponto flutuante, mas a linguagem pode não exigir e o seu real mínimo seria um inteiro:
  - $[+|-] D^+ [.D^+] [E [+|-] D^+]$
  - Inteiros:  $(+|-|\lambda) D^+ = [+|-] D^+$
  - Cadeias:  $' C^* '$  onde  $C$  é ASCII menos  $'$
- (com essa limitação não tratamos os acentos na moda antiga do Pascal para não perder expressividade)

## Comentários em Pascal

- $\{ C^* \}$
- onde  $C$  é ASCII menos  $\}$
- $/* \quad C^* \quad * \quad (* \quad + \quad EC^* \quad * \quad )^* \quad /$
- onde  $C$  é ASCII menos  $*$  e  $E$  é ASCII menos  $/$  e  $*$
- A solução acima vem do AF para comentários (próximo slide)

# AF que reconhece comentários da forma /\* ... \*/

JFLAP : <untitled1>

File Input Test Convert Help

Editor Simulate: /\*CCC\*/

q0 q1 q2 q3 q4

Transitions:  $/$ ,  $*$ ,  $C$ ,  $E$ ,  $*$

Annotations:  
 $C = Vc \text{ diferente de } *$   
 $E = Vc \text{ diferente de } / \text{ e de } *$

q4

/\*CCC\*/

AFD

Step Reset Freeze Thaw Trace Remove

Iniciat JSTOR: Applied Statistics... Wiley InterScience: Jour... Sce185\_2005\_A\_B Microsoft PowerPoint - [...] JFLAP : <untitled1>

12:55

$$/* C^* * (* + EC^* * )^* /$$

JFLAP : <untitled1>

File Input Test Convert Help

Editor Convert FA to RE

Generalized Transition Graph Finished!  
/\*C\*\*(\*+EC\*\*)\*!

Do It Export

```
graph LR; q0((q0)) -- ε --> q0; q4((q4)) -- ε --> q4; q0 -- /*C**(*+EC**)*! --> q4; q4 -- ε --> q0;
```

JFLAP : <untitled2>

File Convert Help

Editor

Edit the regular expression below:  
/\*C\*\*(\*+EC\*\*)\*!

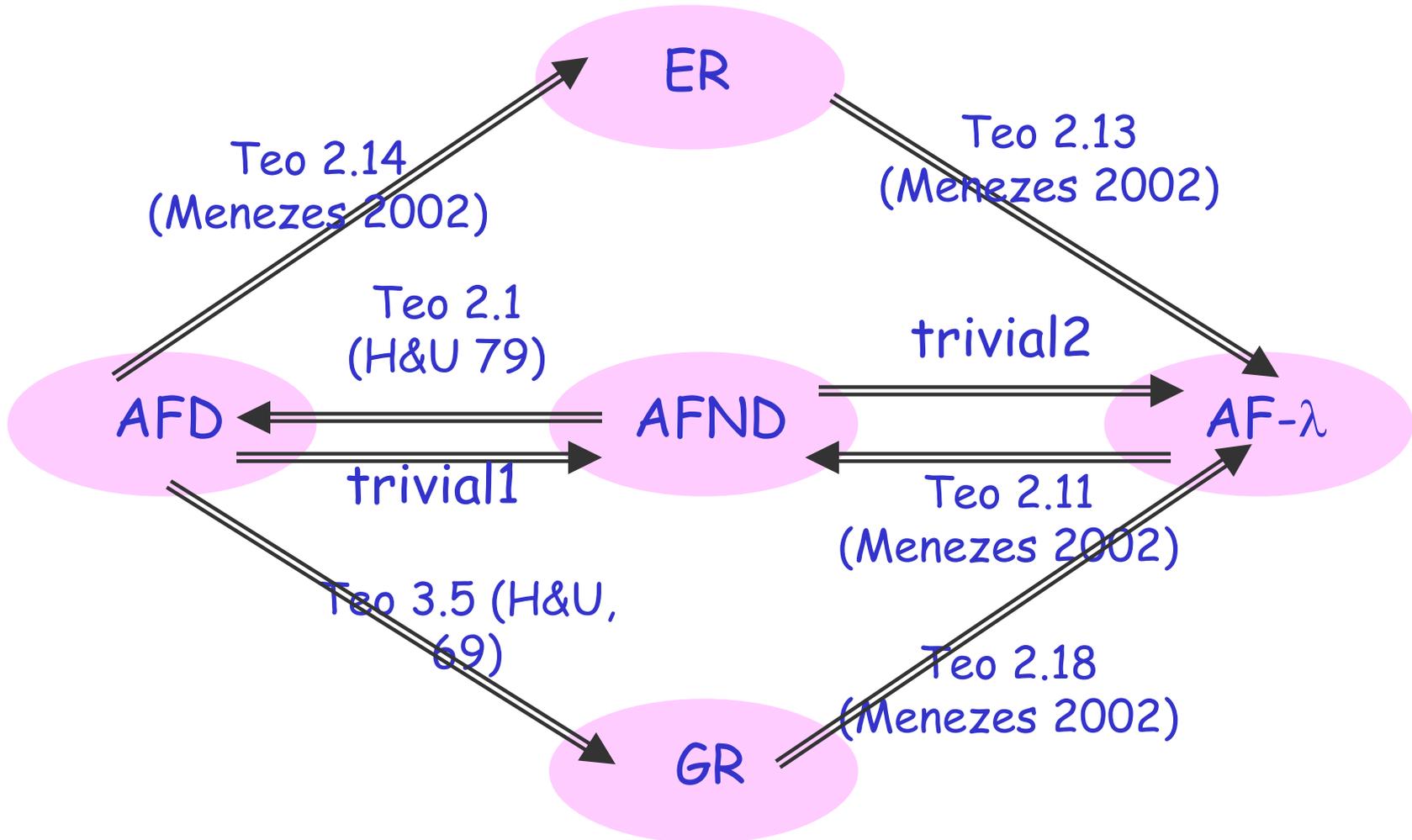
Windows taskbar: Iniciar, JSTOR: Applied Statistics..., Wiley InterScience: Jour..., Sce185\_2005\_A\_B, Microsoft PowerPoint - [...], JFLAP : <untitled1>, JFLAP : <untitled2>, 12:57

# Reais Fortran

- $(+|-|\lambda) (D^+ . | .D^+ | D^+ . D^+) (E (+|-|\lambda) D^+ | \lambda)$
- $Ou$
- $[+|-] (D^+ . | .D^+ | D^+ . D^+) [E [+|-] D^+]$

# Equivalências entre AFD, AFND, AF- $\lambda$ , ER, GR

Trivial2: decorre da definição



Trivial1: colocar { } nos estados