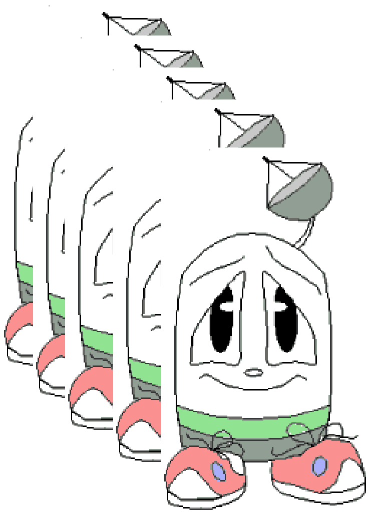
Automatos Finitos



AF Não-determinísticos

Equivalência entre AFDN e AFD

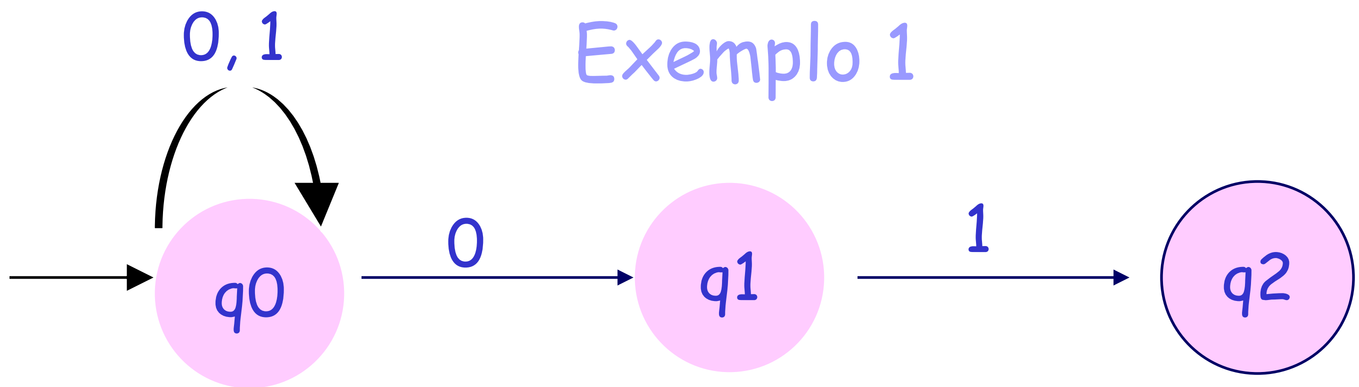
Equivalência entre AF e GR

(H&U, 1969 e 1979), (H;M;U, 2001) e (Menezes, 2002)

AF NÃO-Determinístico (AFND)

- Consideremos uma modificação no modelo do AFD para permitir
  - zero, uma ou mais transições de um estado sobre o MESMO símbolo de entrada.
  - Ou visto de outra forma:
  - a função toma um estado e uma entrada e devolve zero, um ou mais estados.
- Esse modelo é chamado AFND.

Exemplo 1



Esse autômato aceita cadeias de 0's e 1's que terminam em 01.
Analisem a aceitação de "00101"

Quando está em q0 e o símbolo lido é 0 ele tem a opção de:

continuar em q0 no caso do fim da cadeia não estar próximo

OU

ir para q1 porque advinha que o fim está chegando.

E na verdade ele executa as duas opções!

Por isso costumamos pensar que ele "advinha" qual é a alternativa correta de muitas.

AFND

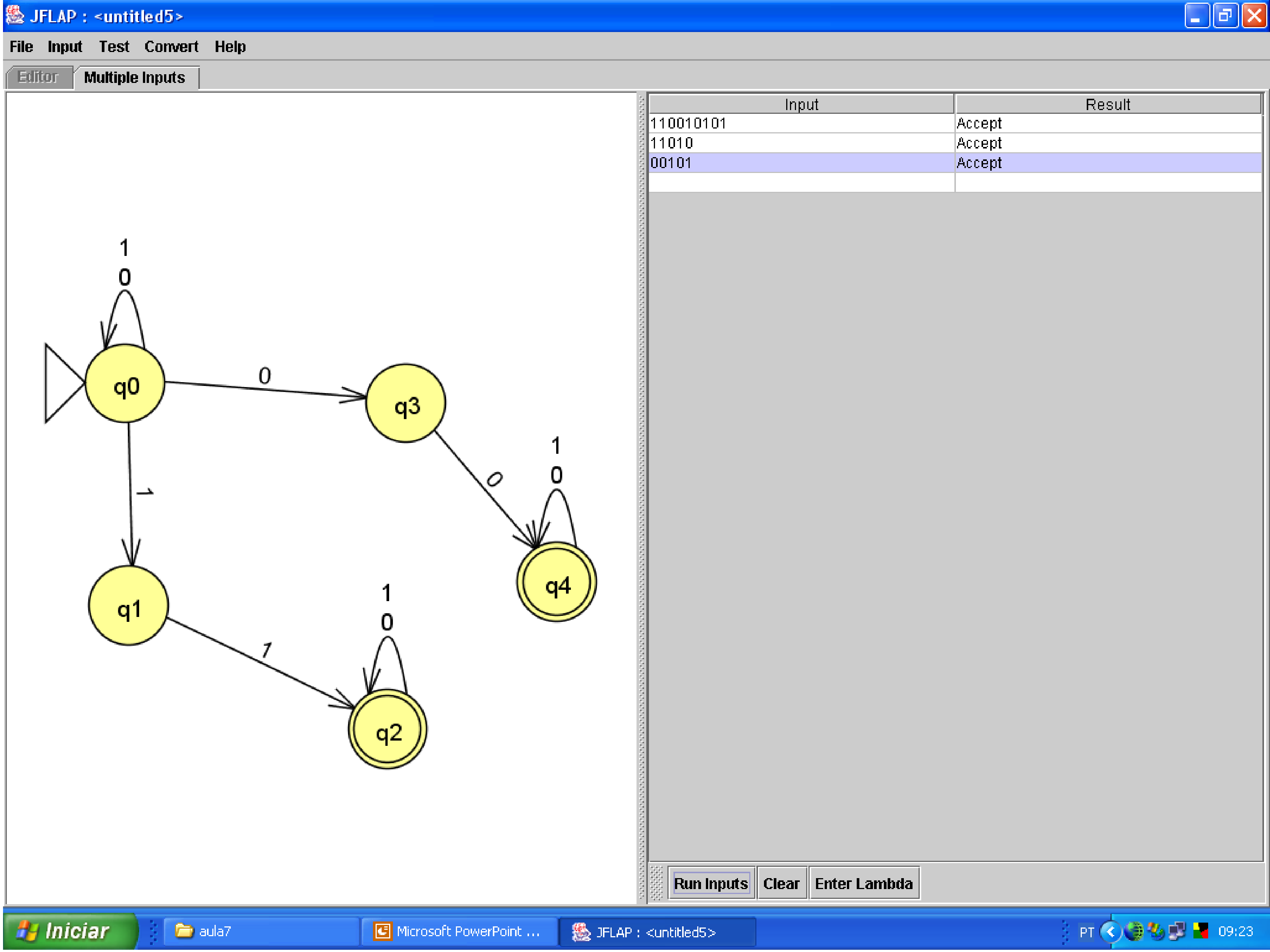
- Uma seqüência de entrada $a_1a_2...a_n$ é aceita por um AFND
  - se existe AO MENOS UMA seqüência de transições, correspondente à seqüência de entrada, que leva o estado inicial para algum estado final.
- Ele funciona como se houvesse a multiplicação da unidade de controle, uma para cada alternativa,
  - processando independentemente, sem compartilhar recursos com as demais!
- Pensem: Será que o não-determinismo aumenta o poder de reconhecimento de linguagens de uma classe de autômatos?

Exemplo 2

- $L(M) = \{x \in \{0,1\}^* \mid x$ tenha dois 0's consecutivos OU dois 1's consecutivos$\}$

O "ou" é não-exclusivo

Ex 2



L(M) = {x ∈{0,1}* | x tenha dois 0's consecutivos OU dois 1's consecutivos}

M = ({q0,q1,q2,q3,q4},{0,1},δ,q0,{q2,q4})

Definição Formal de um AFND

Denotamos um AFND pela 5-tupla $(Q,\Sigma,\delta, q0,F)$
onde Q,Σ, $q0$ e F são os mesmos de um AFD e
δ: $Q \times \Sigma \rightarrow 2^Q$, conjunto potência de Q, isto é, todos os sub-conjuntos de Q
$L(M) = \{w \mid \delta(q0,w)$ contém um estado em $F\}$

No exemplo 2:

	Entrada	
Estado	0	1
q0	{q0,q3}	{q0,q1}
q1	$\varnothing$	{q2}
q2	{q2}	{q2}
q3	{q4}	$\varnothing$
q4	{q4}	{q4}

Exemplo 3

Construir um AFND que aceita cadeias $\in \{1,2,3\}^*$ tal que o último símbolo na cadeia tenha aparecido anteriormente. Por exemplo, 121 é aceita; 31312 não é aceita.

Dica: resolvam para os vocabulários mais simples antes

1) Construir um AFND que aceita cadeias $\in \{1\}^*$ tal que o último símbolo na cadeia tenha aparecido anteriormente
2) Construir um AFND que aceita cadeias $\in \{1,2\}^*$ tal que o último símbolo na cadeia tenha aparecido anteriormente

Equivalência entre AFD e AFND

Teo 2.1 (H&U,79): Seja L o conjunto aceito por um AFND, então existe um AFD que aceita L. Isto é, eles são equivalentes.

Esse teorema responde a primeira pergunta desses slides.

Embora muitas vezes seja mais fácil construir um AFND para uma LR, o AFD tem na prática quase o mesmo número de estados que o AFND, embora ele tenha mais transições.

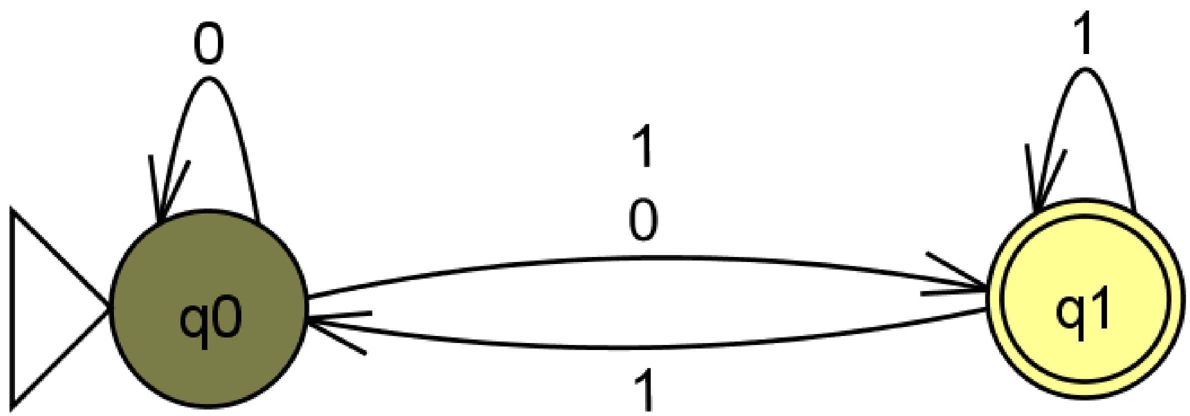
No pior caso, o menor AFD pode ter 2^n estados enquanto que o menor AFND para a mesma linguagem tenha somente n estados.

OBS: ACPD e ACPND (relacionados a GLC) NÃO aceitam a mesma classe de linguagem, já as MTD e MTND (GDC e GEF) são equivalentes.

JFLAP : <untitled7>

File Input Test Convert Help

Editor Simulate: 0111



q0

0111

Exemplo: Seja M = ({q0,q1},{0,1}, δ, q0,{q1})

Es/En	0	1
q0	{q0,q1}	{q1}
q1	∅	{q0,q1}

Step Reset Freeze Thaw Trace Remove

Iniciar aula7 Microsoft PowerPoint ... JFLAP : <untitled7> PT 09:40

- Nós podemos construir um AFD M'= (Q', {0,1}, δ', [q0], F') aceitando L(M) pela construção chamada de "construção de subconjuntos"
  - porque ela envolve a construção de todos os subconjuntos do conjuntos de estados do AFND.

- Ela é tal que Σ é o mesmo do AFND e o estado inicial é o conjunto contendo somente o estado inicial do AFND.

e0 e1 e2

- Q' consiste de todos os subconjuntos de {q0,q1}: [q0], [q1], [q0,q1], e ∅.
  - Note que freqüentemente nem todos os estados são acessíveis do estado inicial e podem ser descartados.

$\delta'([q0],0) = [q0,q1]$ $\delta'([q0],1) = [q1]$

$\delta'([q1],0) = \varnothing$ $\delta'([q1],1) = [q0,q1]$

$\delta'([q0,q1],0) = [q0,q1]$

Pois $\delta(\{q0,q1\},0) = \delta(q0,0) \cup \delta(q1,0) = \{q0,q1\}$

$\{q0,q1\} \cup \varnothing = \{q0,q1\}$

$\delta'([q0,q1],1) = [q0,q1]$

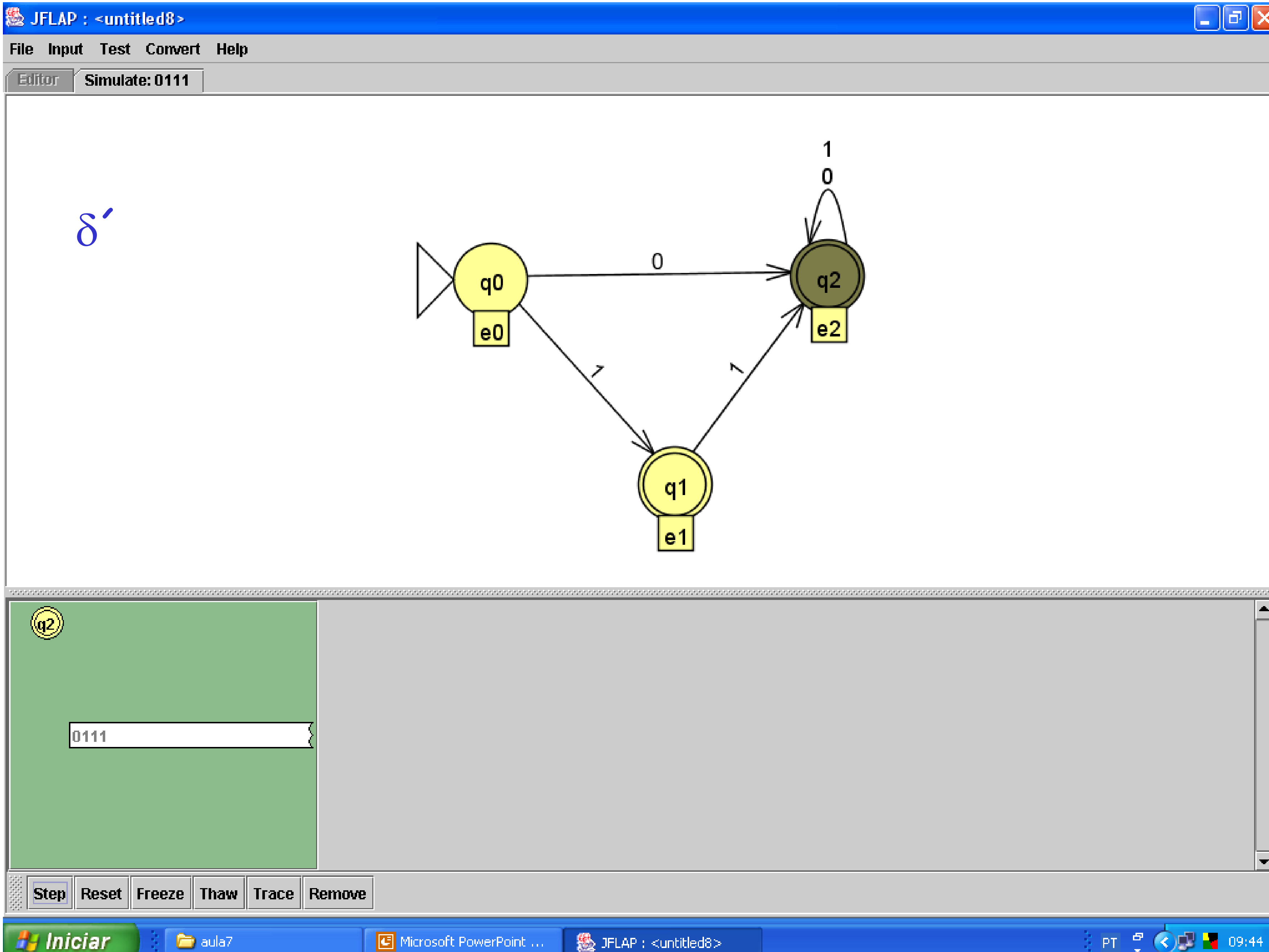
Pois $\delta(\{q0,q1\},1) = \delta(q0,1) \cup \delta(q1,1) =$

$\{q1\} \cup \{q0,q1\} = \{q0,q1\}$

$\delta'(\varnothing,0) = \delta'(\varnothing,1) = \varnothing$

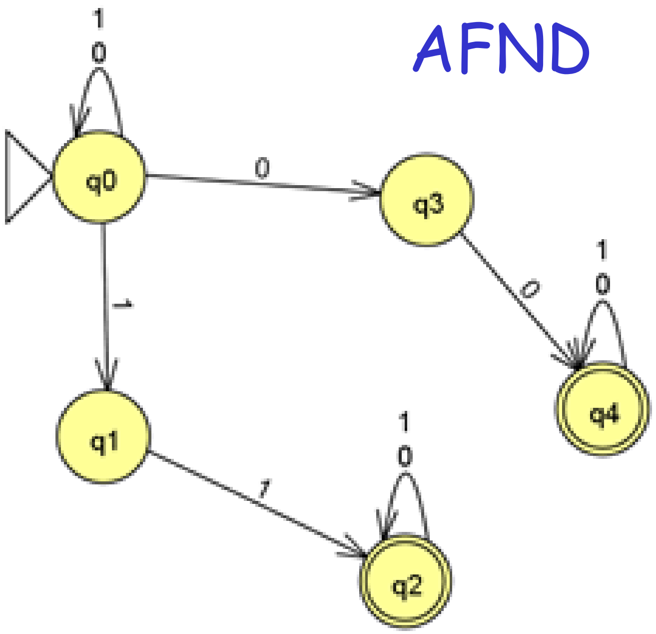
- F' = {[q1],[q0,q1]} isto é, estados onde F antigo estava presente

$$M' = (\{e0, e1, e2\}, \{0,1\}, \delta', e0, \{e1, e2\})$$



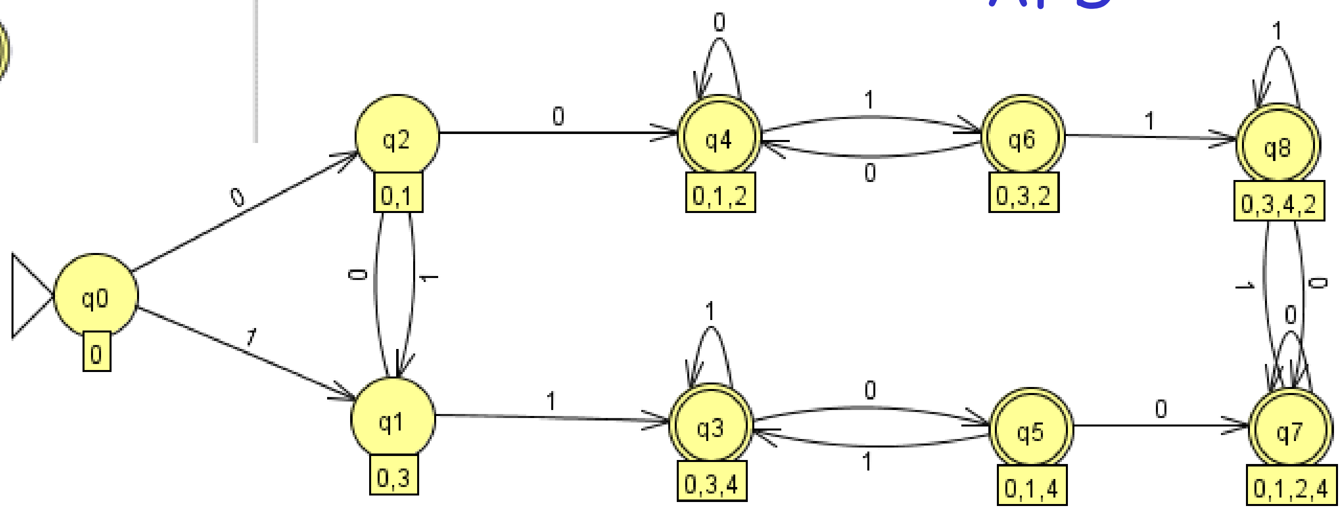
$L(M) = \{x \in \{0,1\}^* \mid x$ tenha dois 0's consecutivos OU dois 1's consecutivos$\}$

AFND



Transformação realizada pelo JFlap

AFD



Estados de não-aceitação em AFD

(H,M,U, 2001) pg 67

- A definição estrita de um AFD EXIGE que todo estado tenha uma transição para cada símbolo de entrada.

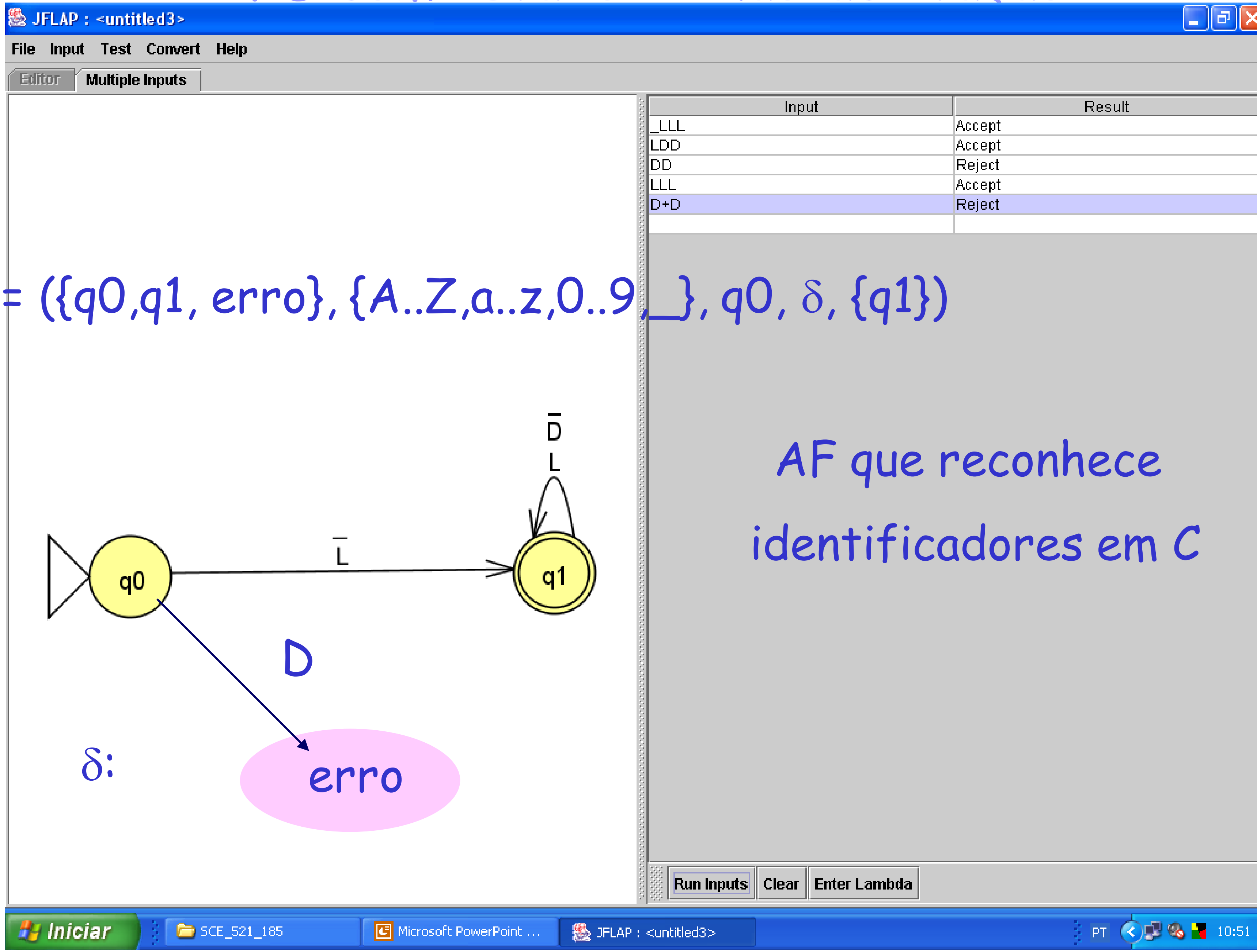
- Se seguirmos esta definição o exemplo de AFD na seção de equivalência não é AFD no senso estrito.

- Porém, temos a facilidade de criar um estado de não-aceitação (erro) para o qual podem ir todas as entradas não necessárias na definição de uma linguagem.

- Por esta razão, relaxamos esta exigência e dizemos que um AFD tem NO MÁXIMO UMA transição em cada estado para cada símbolo em vez de exatamente uma transição.

AFD com estado de não-aceitação

$M = (\{q0, q1, erro\}, \{A..Z, a..z, 0..9, \_\}, q0, \delta, \{q1\})$



AF que reconhece identificadores em C

Equivalência entre AFD e GR

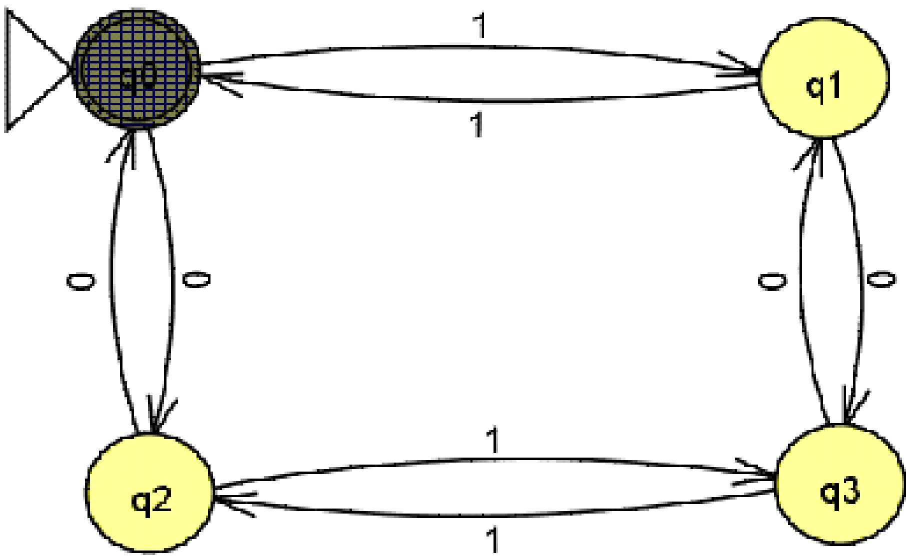
- Teo 3.5 (H&U, 69) Dado um AF M, existe uma gramática do tipo 3 tal que L(G) = L(M).
- Prova: Sem perda de generalidade seja
  $M = (K, \Sigma, \delta, qo, F)$ um AFD. Definimos um GR = (K, Σ, P, qo):

1. B -> aC está em P se $\delta(B,a) = C$
2. B -> a está em P se $\delta(B,a) = C$ e C está em F.

- Se qo está em F então λ está em T(M). Neste caso, L(G) = T(M) - λ.
- Pelo Teo 2.1 (H&U, 69) podemos obter a partir de G uma nova GR onde:

$$L(G1) = L(G) \cup \lambda = T(M)$$

Exemplo: L(M) = {w | w possui um nro par de 0´s e de 1´s }



Como q0 é final e aparece do lado direito, temos que acrescentar λ à L(G) com as regras:

S->0Q2, S->1Q1, S-> λ

E

G1=({S,Q0,Q1,Q2,Q3},{0,1},P1,S)

- G = ({Q0,Q1,Q2,Q3},{0,1},P,Q0)

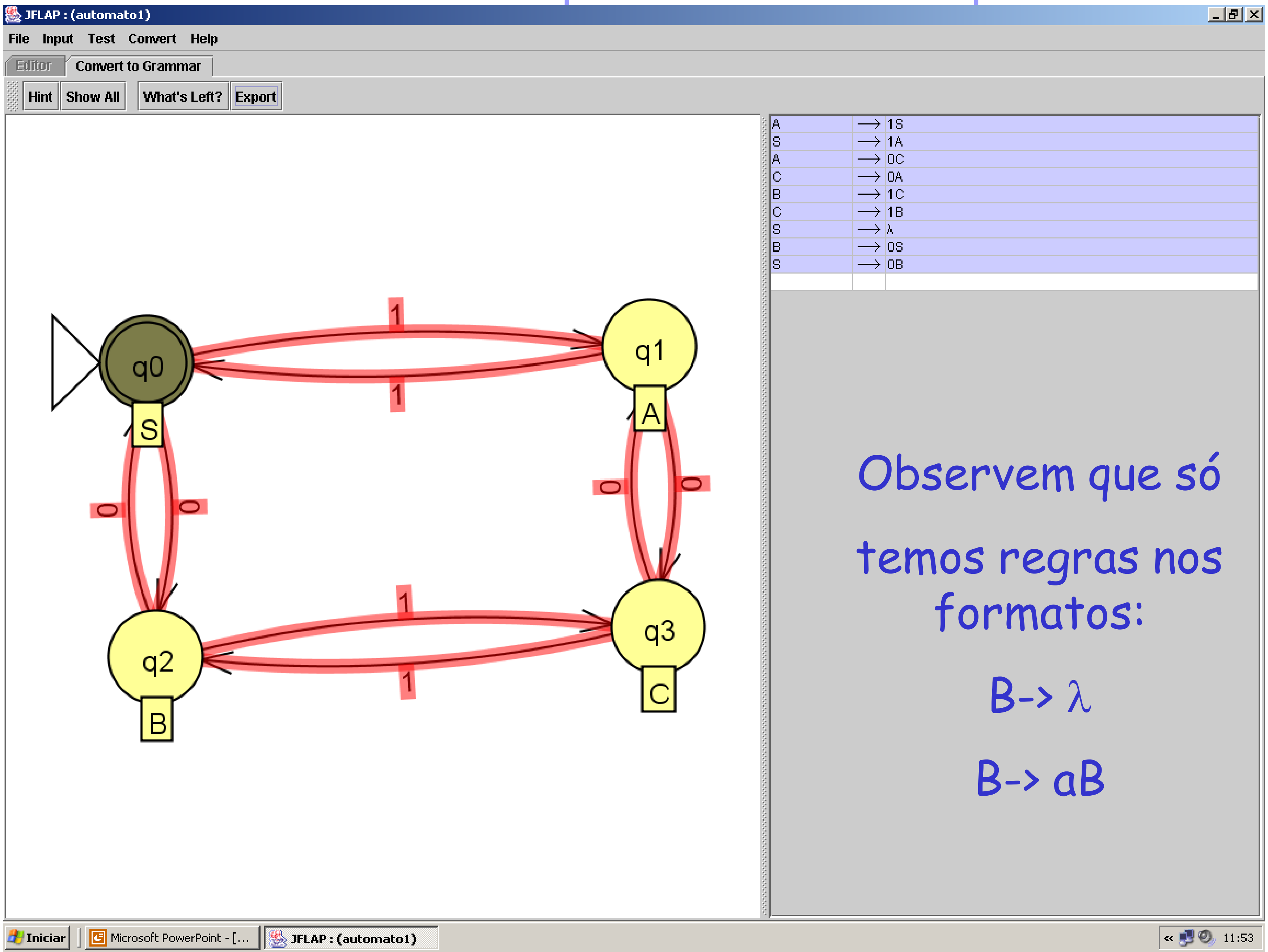
- δ(qo,0) = q2   Qo ->0Q2 δ(qo,1) = q1 Q0->1Q1
- δ(q1,0) = q3   Q1->0Q3 δ(q1,1) = q0 Q1->1Q0
-    Q1->1
- δ(q2,0) = q0 Q2->0Q0 δ(q2,1) = q3 Q2->1Q3
-    Q2->0
- δ(q3,0) = q1 Q3->0Q1 δ(q3,1) = q2 Q3->1Q2

JFLAP e a conversão AF -> GR

- O algoritmo que o JFLAP usa é diferente do Teo 3.5 visto,
  - pois permite que a cadeia nula apareça numa regra da GR sem que seja, unicamente, para derivar a cadeia nula quando esta pertence à linguagem.

- Isto quer dizer que a concepção de GR para o JFLAP é mais flexível do que a vista em sala.

Ex1: Usando o JFLAP para converter AF para GR

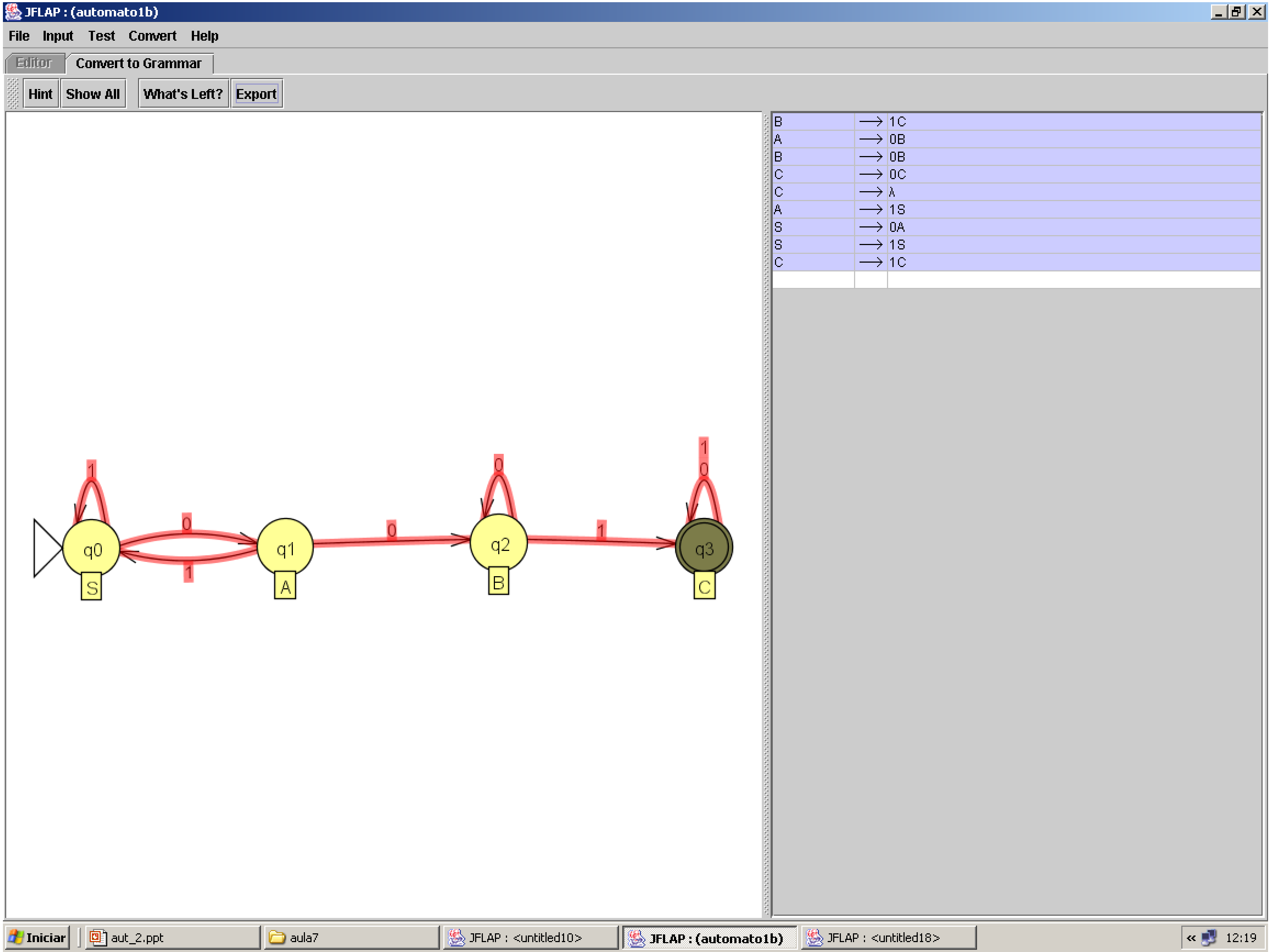


Observem que só temos regras nos formatos:

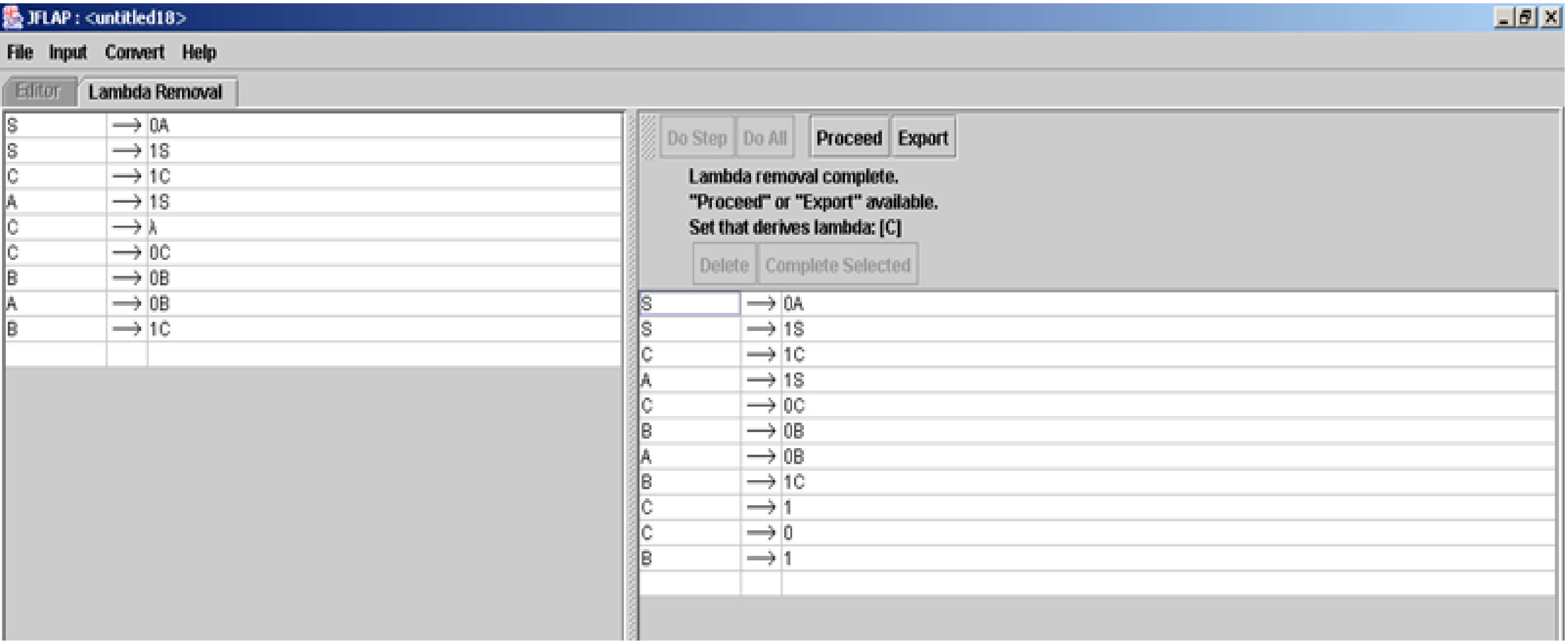
B-> λ

B-> aB

Ex2: Usando o JFLAP e transformações de gramáticas



Remoção de regras com cadeia nula do Ex2

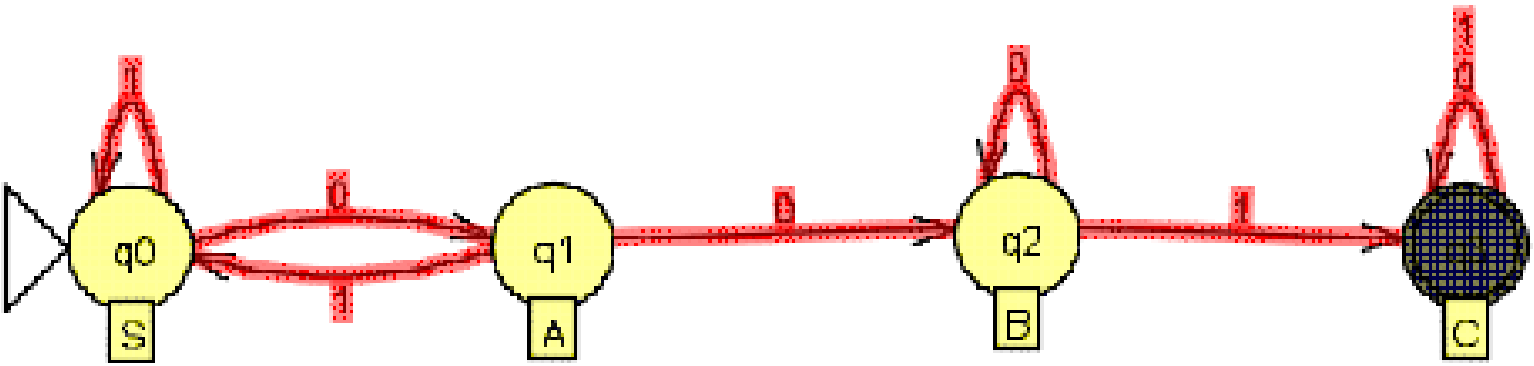


Como o símbolo inicial não leva em λ

Podemos remover ele da gramática

Exercício 1

- Usem o Teo 3.5 com o Ex2 para comparar com a saída do JFLAP após a remoção da cadeia nula.



Exercício 2

1. Fazer um AFD M que reconhece:
   a) $L(M) = \{x \in \{0,1\}^* \mid$ nro de 1´s em x é multiplo de 3$\}$
   b) $L(M) = \{x \in \{0,1\}^* \mid$ x contém a subcadeia 001$\}$

2. Fazer um AFD M que reconhece:
   a) a formação de identificadores em Pascal
   b) números inteiros sem sinal em Pascal
   c) Operadores relacionais em Pascal

3. Para $\Sigma = \{0,1\}$ faça AFD´s que reconheçam $L1 = \emptyset$ e $L2 = \Sigma^*$

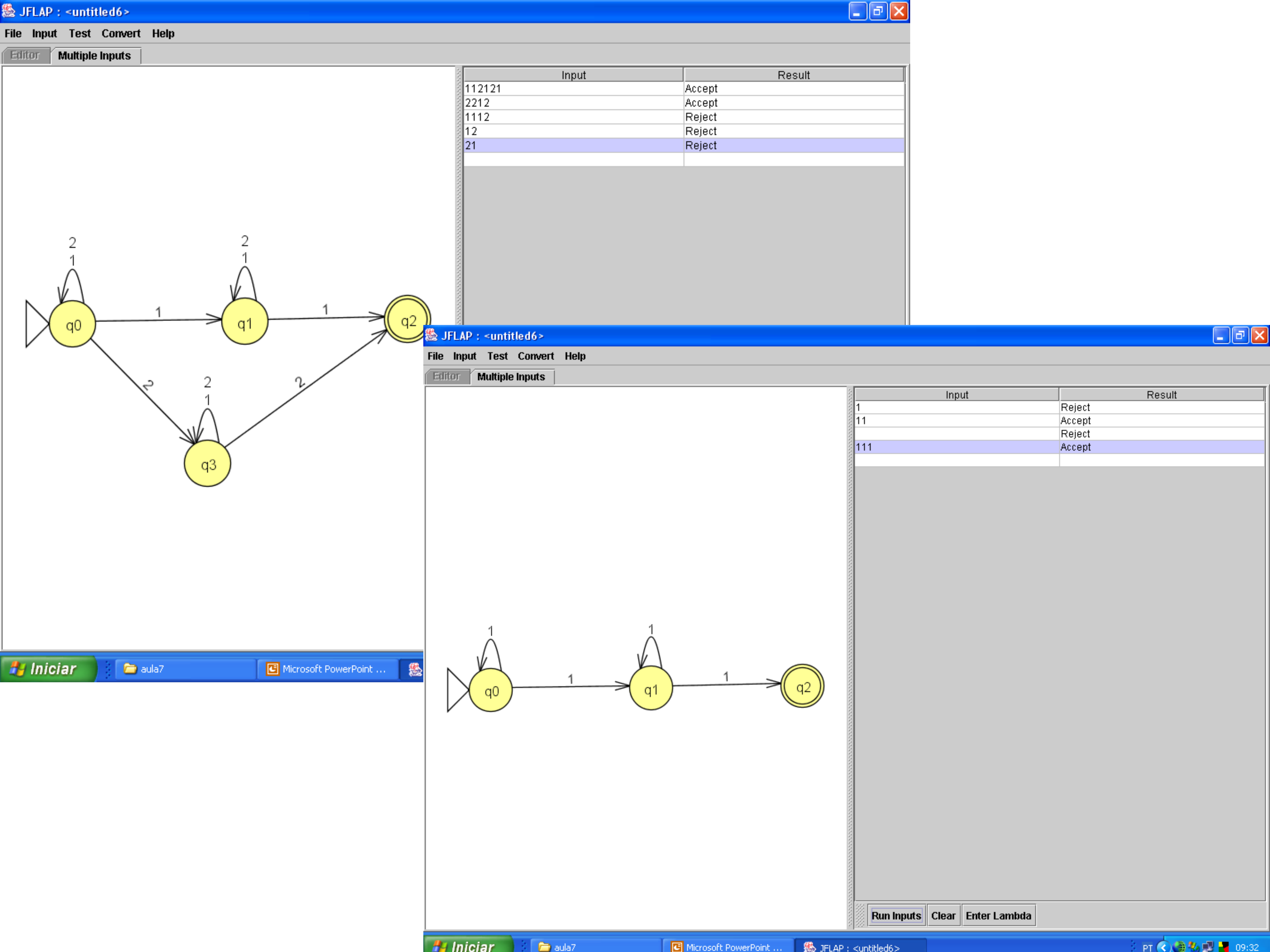
4. Construa um AFD M que reconhece $L(M) = \{ab^nc \mid n >= 0\}$

5. Construa um AFD M que reconhece $L(M) =\{a^nb^m \mid n,m >= 0$ e $n+m > 0\}$. Veja que $n+m > 0$ implica na não possibilidade da vazia nula.

6. Construa o AF e depois a GR para $L1 = \{a^nb^m \mid n,m >= 0\}$ e $L2 = \{a,b\}^*$

Solução do Exemplo 3

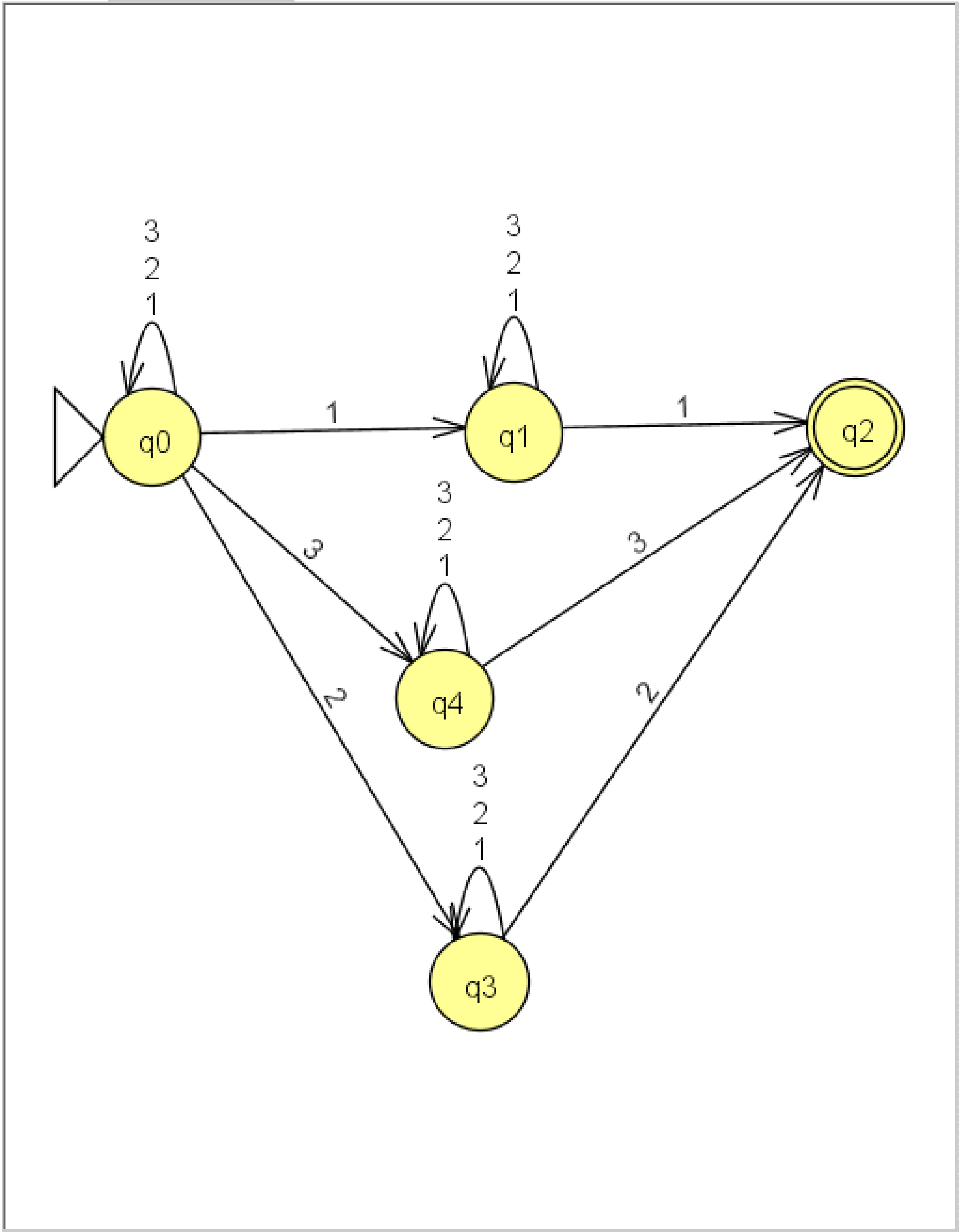
Construir um AFND que aceita cadeias $\in \{1,2,3\}^*$ tal que o último símbolo na cadeia tenha aparecido anteriormente. Por exemplo, 121 é aceita; 31312 não é aceita.


JFLAP : <untitled6>
File Input Test Convert Help
Editor Multiple Inputs
Input Result
112121 Accept
2212 Accept
1112 Reject
12 Reject
21 Reject
q0 q1 q2 q3
1 2
Iniciar aula7 Microsoft PowerPoint ...
JFLAP : <untitled6>
File Input Test Convert Help
Editor Multiple Inputs
Input Result
1 Reject
11 Accept
Reject
111 Accept
q0 q1 q2
Run Inputs Clear Enter Lambda
Iniciar aula7 Microsoft PowerPoint ... JFLAP : <untitled6>
PT 09:32

JFLAP : <untitled6>

File Input Test Convert Help

Editor | Multiple Inputs



Input	Result
121	Accept
31312	Reject
123	Reject

Run Inputs | Clear | Enter Lambda

Iniciar | aula7 | Microsoft PowerPoint ... | JFLAP : <untitled6> | PT | 09:36