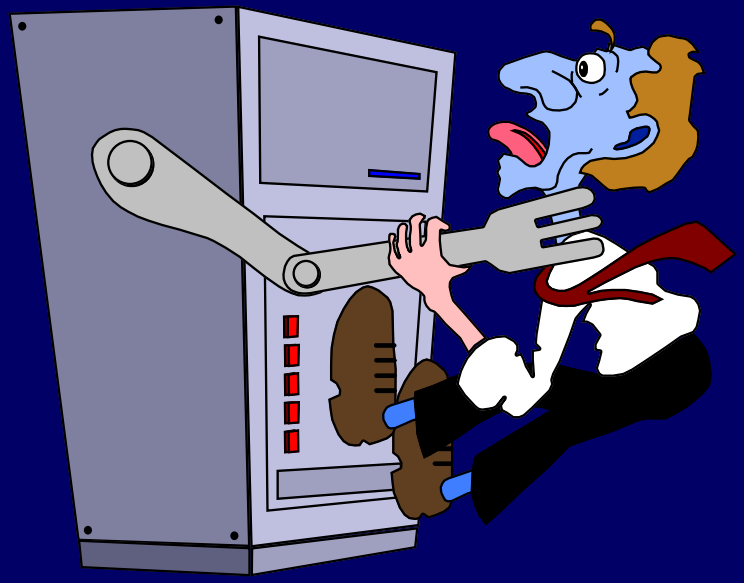
Máquinas de Turing 3



Exercícios
Máquinas de Turing com Múltiplas Fitas
Máquinas de Turing Não-deterministicas
A Tese/Hipótese de Church-Turing
Linguagens decidíveis por Máquinas de Turing (Recursivas)
Linguagens Aceitas/Reconhecidas por Máquinas de Turing (Recursivamente Enumeráveis)

Usos de uma MT

- como reconhecedor de linguagens (Visto)
- para calcular funções
- para processar problemas de decisão (procedimento)

MT como um processador de funções inteiras

- Tradicionalmente, os inteiros são representados em unário

- O inteiro i >= 0 é representado pela cadeia 0^{i}.
- Deve-se estabelecer um mapeamento: zero é representado por 0; 1 por 00,.../ se naturais 1 = 0; 2 = 11, 3 = 111, ...

- Se a função tem k argumentos (i1, i2, ..., ik) então esses inteiros são colocados na fita separados por 1´s como:

$$0^{i1}\ 1\ 0^{i2}\ 1\ \ldots\ 1\ 0^{ik}$$

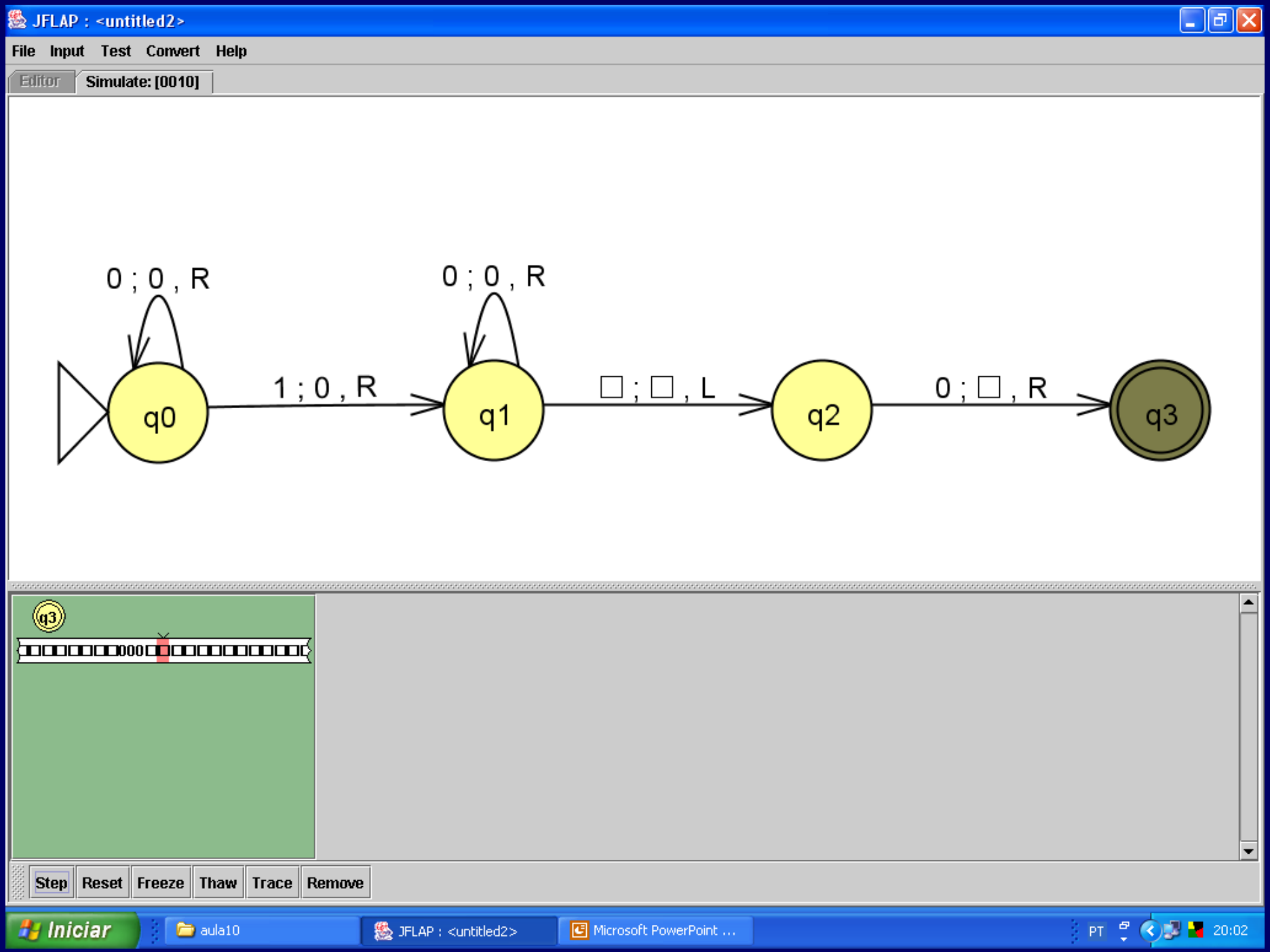
- O inverso também é possível.
- Se a máquina pára (não importa se num estado final) com a fita consistindo de 0^{m} para algum m então dizemos que f(i1,i2,...ik) = m, onde f é uma função de k argumentos computados por essa MT.

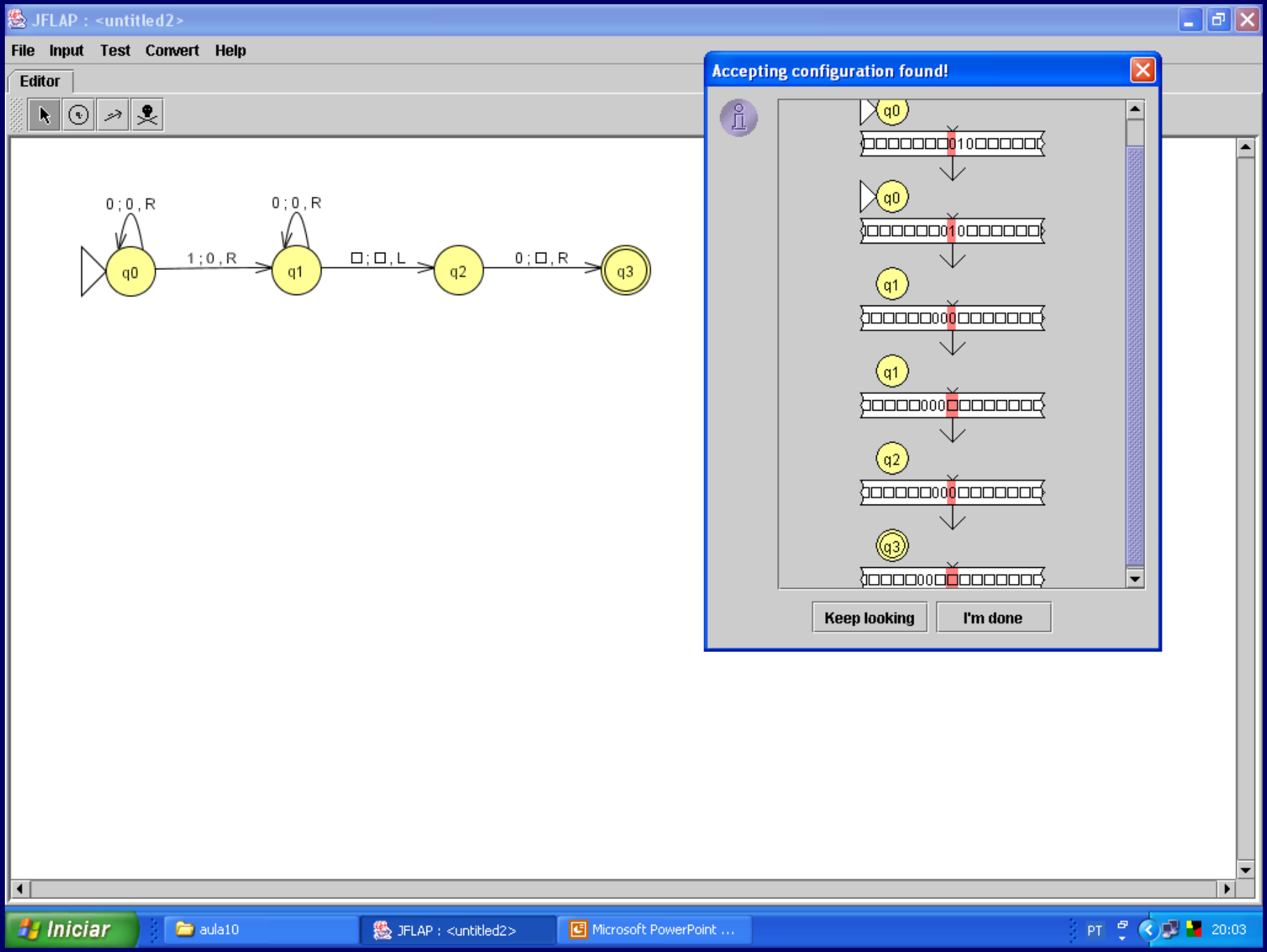
MT que soma dois números naturais

- Consider the addition function $a+b$ on non-negative integers
- Encode the input string on the tape as 0^a10^b
- The Turing machine will halt with 0^{a+b} on the tape

- Processo:
- Read 0's and rewrite them on the tape moving right until the 1 is read
- Replace the 1 with a 0 (SHIFT), and continue moving right without changing the tape until a blank is found
- Move left and replace the 0 with a blank

- Não há zeros, então o 1 = 0; 2 = 00, ...


JFLAP : <untitled2>
File Input Test Convert Help
Editor Simulate: [0010]
0 ; 0 , R
0 ; 0 , R
1 ; 0 , R
□ ; □ , L
0 ; □ , R
q0
q1
q2
q3
q3
Step Reset Freeze Thaw Trace Remove
Iniciar
aula10
JFLAP : <untitled2>
Microsoft PowerPoint ...
PT
20:02


JFLAP : <untitled2>
File Input Test Convert Help
Editor
0 ; 0 , R
0 ; 0 , R
1 ; 0 , R
□ ; □ , L
0 ; □ , R
q0
q1
q2
q3
Accepting configuration found!
Keep looking
I'm done
Iniciar
aula10
JFLAP : <untitled2>
Microsoft PowerPoint ...
PT
20:03

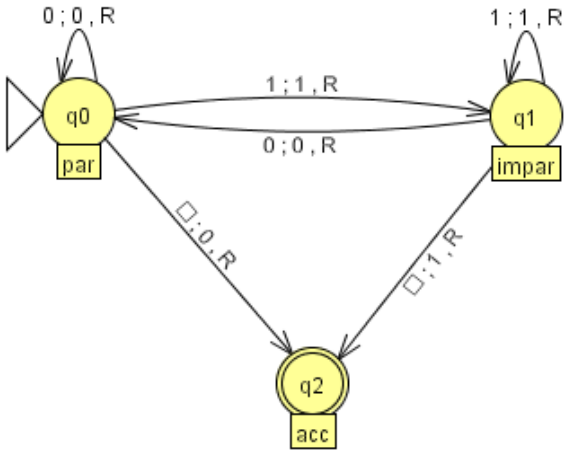
MT para processar problemas de decisão

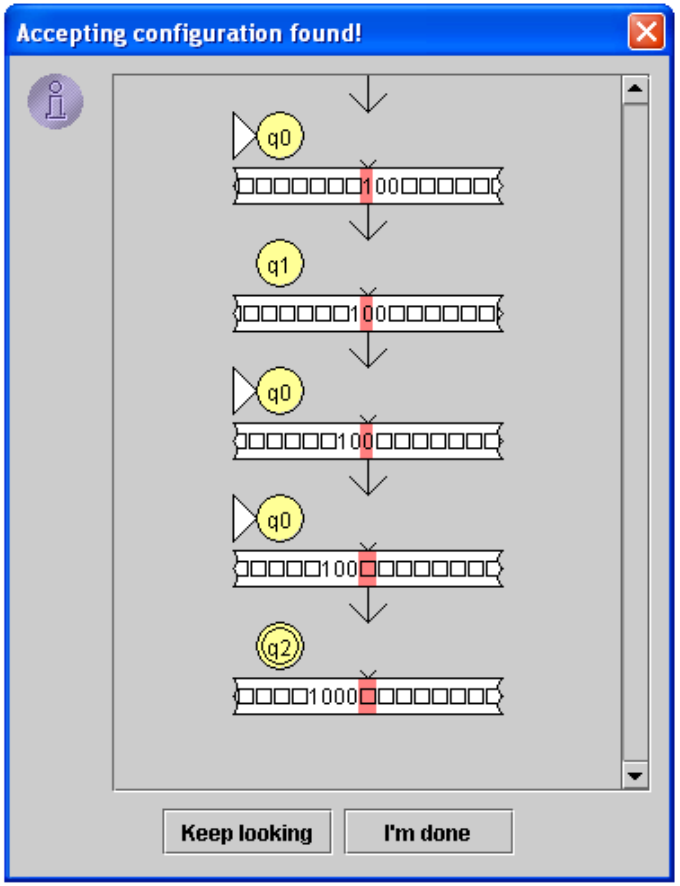
- Quando usamos a MT para decidir (responder T/F) alguma propriedade, depois que a máquina parar, olhamos na fita a procura do resultado 0/1 após a cadeia de entrada.

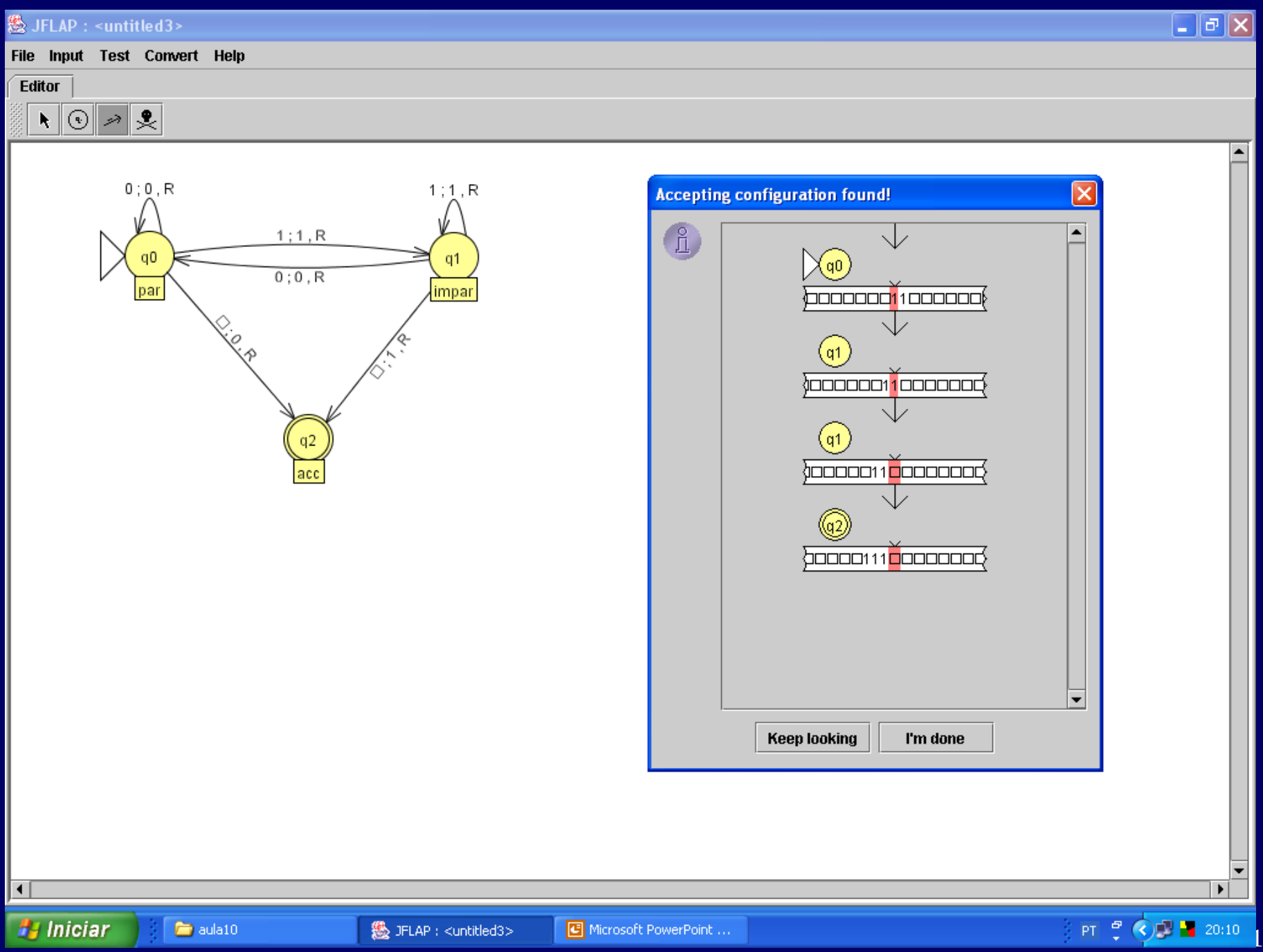
MT que decide se um número binário é par ou impar

- Fica em q0 enquanto par
- Fica em q1 enquanto impar

- Se Par - pára e escreve 0
- Se Impar - pára e escreve 1


0 ; 0 , R
1 ; 1 , R
q0
par
1 ; 1 , R
0 ; 0 , R
q1
impar
□ ; 0 , R
□ ; 1 , R
q2
acc


Accepting configuration found!
q0
q1
q0
q0
q2
Keep looking
I'm done


JFLAP : <untitled3>
File Input Test Convert Help
Editor
0 ; 0 , R
1 ; 1 , R
q0
par
1 ; 1 , R
0 ; 0 , R
q1
impar
□ ; 0 , R
□ ; 1 , R
q2
acc
Accepting configuration found!
q0
11
q1
11
q1
11
q2
111
Keep looking
I'm done
Iniciar
aula10
JFLAP : <untitled3>
Microsoft PowerPoint ...
PT
20:10

Exercícios

1) Construir uma MT que decida se uma seqüência de parênteses é bem formada.
   - Escreve 0 se mal formada
   - Escreve 1 se bem formada

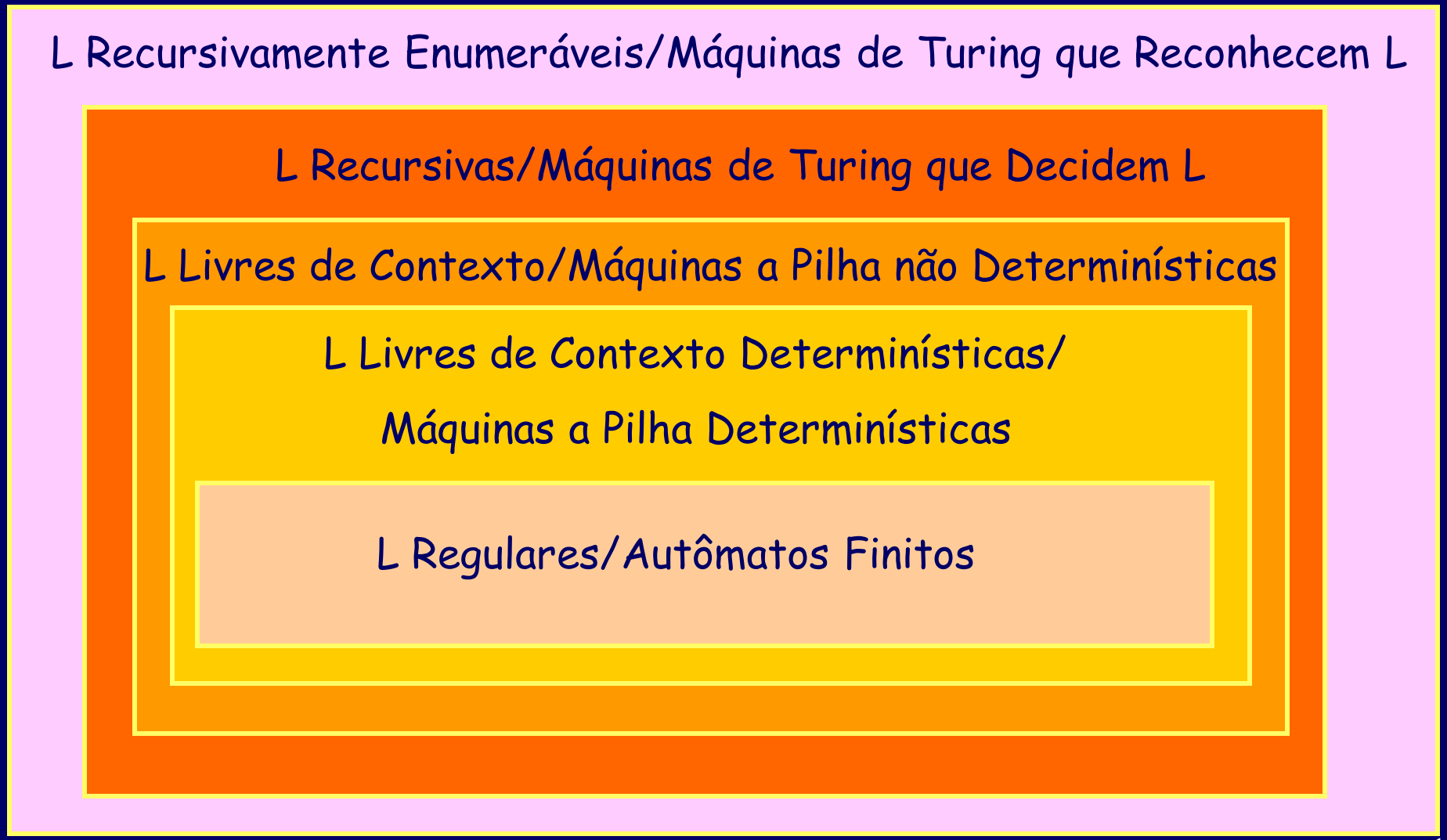
- Dica: Use um ALL; considere que a cadeia de parênteses é limitada por < e > (um a esq e outra à direita).
- Idéia: Procurem por um) e substitua por X e em seguida voltar a esquerda procurando o (mais próximo para substituir por X também.

2) Construir uma MT que dada uma cadeia w pertencente ao fecho de {0,1} duplique w. Quando a máquina parar a fita deve conter w#w sendo que # indica fim de w.

Lembrem que:

- MT para calcular funções:
  - Se f(n)
    - não é definida para todo n → f(n) é função parcial;
    - se sempre é definida é função total
  - Se MT computa f(n)
    - para todo n → f(n) é recursiva (algoritmo que computa f(n))
    - sempre que f(n) é definida mas NÃO pára para aqueles n para os quais f(n) não é definida → f(n) é parcialmente recursiva (ou recursivamente enumerável)
- MT para processar problemas de decisão:
  - Quando consideramos a MT como procedimento, se ela pára para toda a entrada dizemos que o procedimento é um algoritmo.

Hierarquia das Classes de Máquinas e Linguagens



Modelos Equivalentes

- Máquinas de Turing Determinísticas (MTD) são extremamente poderosas. Nós podemos simular outros modelos com elas e vice-versa.

- As variações são equivalentes, isto é, reconhecem a mesma classe de linguagens (seja ela recursiva/re).

- Uma variação simples é a MT que tem a habilidade de permanecer parada após a leitura.
  - Como provamos que esta variação tem poder equivalente ao da MTD?
  - Lembrem que para mostrar que 2 modelos são equivalentes mostramos que simulamos um pelo outro.

- Vamos ver outro exemplo em que a simulação ocorre com somente uma perda polinomial de eficiência

Máquinas de Turing com múltiplas fitas

SIP 136-138

Inicialmente a entrada é escrita na primeira fita. As outras começam com branco.

As transições permitem acesso simultâneo a todas as fitas:

δ: Q X Γ^k -> Q X Γ^k X {L,R}^{k} Ex: δ(qi,a1,...,ak) = (qj,b1,b2, ..., bk, L,R,L,R...)

Robustez

- Máquinas de Turing com múltiplas fitas são polinomialmente equivalentes a máquinas com uma fita (que são casos especiais da primeira).

- Teo 3.8 (Sipser) Toda MT com múltiplas fitas tem uma MT equivalente com uma única fita.
  - A idéia é mostrar como simular M com k fitas em S com 1 fita.
  - S simula o efeito das k fitas armazenando suas informações uma sua fita, separando-as com # e marca as posições de suas cabeças com símbolos marcados (a. e b., por exemplo que são acrescidos ao alfabeto da fita):

    #w1.w2…wn# .# .# ….#
  - O que fazer quando a cabeça chega num #?

- Nós podemos agora fazer uma afirmação muito mais forte com respeito a robustez de Máquinas de Turing:

A Hipótese de Church-Turing

Noção Intuitiva de algorítmo ≡ Máquina de Turing que decide

Lembrem que alguns problemas são parcialmente decidíveis, isto é, existe uma máquina de Turing que os reconhece

Existem mais modelos Equivalentes?

Vamos ver um modelo computacional menos realista - MT Não-Determinísticas -

que pode ser simulado com uma MT Determinística com uma perda exponencial de eficiência.

Computações

$\delta(q,X) = \{(q1,Y1,D1),\ldots,(qk,Yk,Dk)\}$

Descrição Alternativa

- Uma máquina não-determinística tem dois estágios:
- um de advinhação da resposta e
- outro de verificação.

- É um modelo para capturar a noção de verificação em tempo polinomial (NP) e não um método para resolver problemas de decisão.

Exemplos

(1) Uma MT não-determinística que checa se 2 vértices são conectados em um grafo simplesmente "chuta" um caminho entre eles. Após isto a MT necessita somente verificar se o caminho é válido.

Lembram que vimos 2 problemas no início do curso:

- O problema da Mesa (CICLO-HAMILTONIANO)
- O problema do Pacote (CAMINHO-HAMILTONIANO)

(2) Para o problema de verificar se um grafo é hamiltoniano a MT "chuta" a lista de vértices no ciclo hamiltoniano. Se o grafo for hamiltoniano o próprio ciclo oferece toda a informação para verificar este fato. Se o grafo não for nenhuma lista de vértices vai enganar o verificador que facilmente verifica se a lista é válida.

Sumário

- Apresentamos os dois modelos principais de computação: MT Determinística e MT Não-determinística.
- A MTD pode ser simulada por uma MTND com uma perda exponencial de eficiência.
- Por isto que dizemos que a classe NP é a classe de todos os problemas de decisão "solúveis" por algoritmos não-determinísticos de tempo polinomial.
- N = não-determinístico P = polinomial
- A verificação é em tempo polinomial pois não contamos o tempo de advinhação, que será exponencial!

Sumário

- A Hipótese de Church-Turing: MT Deterministicas que decidem são equivalentes a nossa noção intuitiva de algorítmos.

- Sabendo disto, podemos descrever usualmente algoritmos em pseudo-código ao invés de escrever em MT.