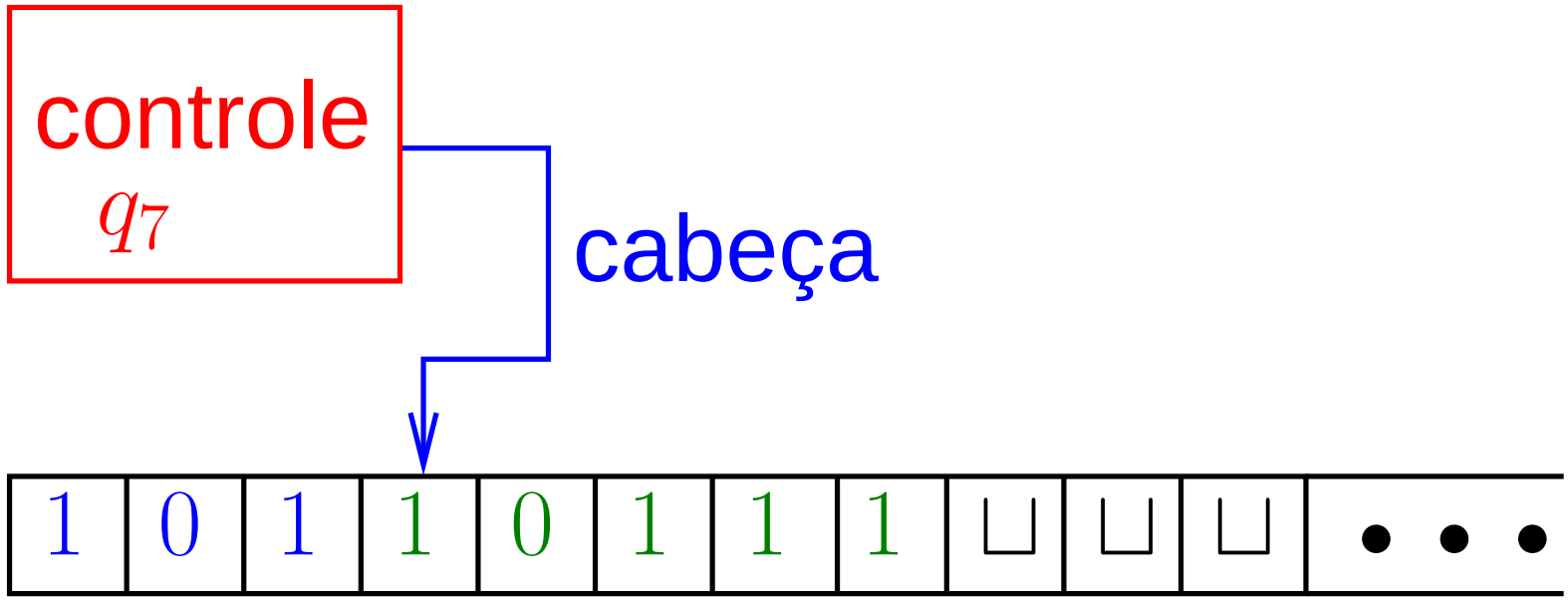
Melhores momentos

AULA PASSADA

Configurações



fita de leitura e escrita

Configuração $1\,0\,1\,q_7\,1\,0\,1\,1\,1$

Linguagem de uma MT

Uma máquina de Turing **aceita** a entrada w se existe uma sequência de configurações $C_1, C_2, \ldots, C_k$ tal que

1. C_1 é a configuração inicial de M com entrada w,
2. cada C_i leva a C_{i+1} para $i = 1, \ldots, k-1$, e
3. C_k é uma configuração de aceitação.

A **linguagem de** M, ou a **linguagem reconhecida** por M, denotada por $L(M)$, é o conjunto de cadeias que ela aceita.

Liguagens reconhecíveis

Uma linguagem é **Turing-reconhecível** se existe alguma máquina de Turing que a reconhece (linguagem recursivamente enumerável).

Exemplos:
As linguagens a seguir são Turing-reconhecíveis:

- $L(M_0) = \{u\,0\,0 : u \in \{0,1\}^*\}$
- $L(M_2) = \{0^{2^n} : n \geq 0\}$

Observação: se w não está em $L(M)$ então M pode **não parar** com w como entrada.

Liguagens reconhecíveis e decidíveis

Uma máquina de Turing M é **decisora** se sempre aceita ou rejeita sua entrada. Dizemos que M **decide** $L(M)$.

Uma linguagem é **Turing-decidível** ou simplesmente **decidível** se existe alguma máquina de Turing que a decide (linguagem recursiva).

Exemplos:
As linguagens a seguir são Turing-decidíveis:

- $L(M_0) = \{u\,0\,0 : u \in \{0,1\}^*\}$
- $L(M_2) = \{0^{2^n} : n \geq 0\}$

Toda linguagem decidível é Turing-reconhecível.

AULA 4

Variantes de Máquinas de Turing

MS 3.2

Variante simples

$$\delta : (Q \setminus \{q_{\text{aceitação}}, q_{\text{rejeição}}\}) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, P\},$$

onde P representa que a cabeça deve ficar parada.

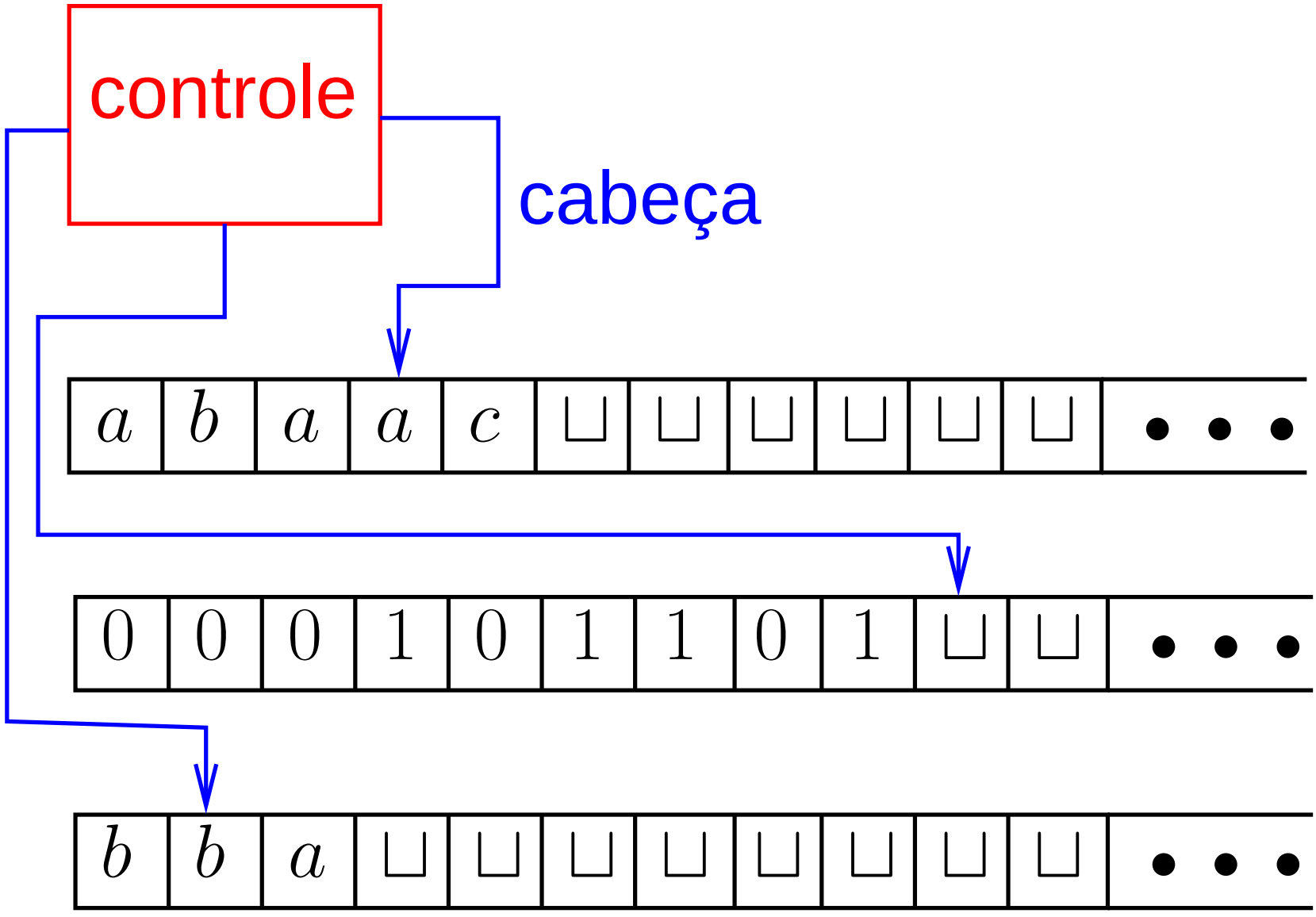
Variante simples

$$\delta : (Q \setminus \{q_{\text{aceitação}}, q_{\text{rejeição}}\}) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, P\},$$

onde P representa que a cabeça deve ficar parada.

Uma MT com essa possibilidade pode ser simulada por uma MT comum trocando cada transição que não se move por duas transições: uma para a direita e outra para a esquerda.

Máquinas de Turing multifita



Sobre os componentes

Componentes de uma máquina de Turing multifica:

Controle: possui um conjunto finito de estados, dentres eles há 3 estados especiais: *inicial*, *aceitação* e *rejeição*. Inicialmente está no estado inicial.

Cabeças: Uma cabeça para cada fita. Cada cabeça lê o símbolo na posição sobre a qual está, escreve um símbolo nessa posição e move uma posição para a direita ou esquerda. Inicialmente cada cabeça está sobre a primeira posição de sua fita;

Fitas: infinitas à direita. Inicialmente a primeira fita contém a cadeia representando a entrada do problema nas primeiras posições, as demais posições contém ␣s. As demais fitas contém apenas ␣s.

Máquinas de Turing multifita

Uma **máquina de Turing multifita** é como uma MT comum com várias fitas.

Cada fita tem sua própria cabeça para leitura e escrita.

Inicialmente a entrada aparece na fita 1 e as outras fitas estão em branco.

Função de transição para máquina com k fitas:

$$\delta : (Q \setminus \{q_{\text{aceitação}}, q_{\text{rejeição}}\}) \times \Gamma^k \longrightarrow Q \times \Gamma^k \times \{L, R, P\}^k.$$

Máquinas de Turing multifita

A expressão

$$\delta(q_i, a_1, \ldots, a_k) = (q_j, b_1, \ldots, b_k, L, R, \ldots, L)$$

significa que se a máquina está no estado q_i e as cabeças estão lendo os símbolos

$$a_1, a_2, \ldots, a_k,$$

respectivamente nas fitas de 1 a k, ela vai para o estado q_j e escreve

$$b_1, b_2, \ldots, b_k,$$

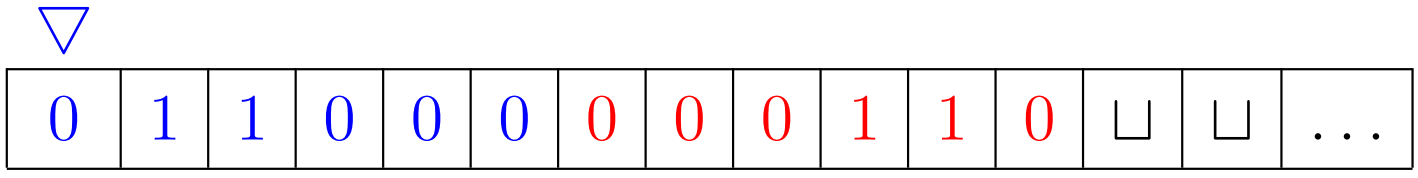
nas k-fitas, movendo a primeira cabeça para a esquerda, a segunda para a direita, $\ldots$, e a última para a esquerda.

Exemplo de MT multifita

Descrição alto nível de uma máquina de Turing que decide se uma dada cadeia w está na linguagem

$$\{z : z \in \{0,1\}^*,\ z = z^R\}.$$

$M_1 =$ "Com entrada w:



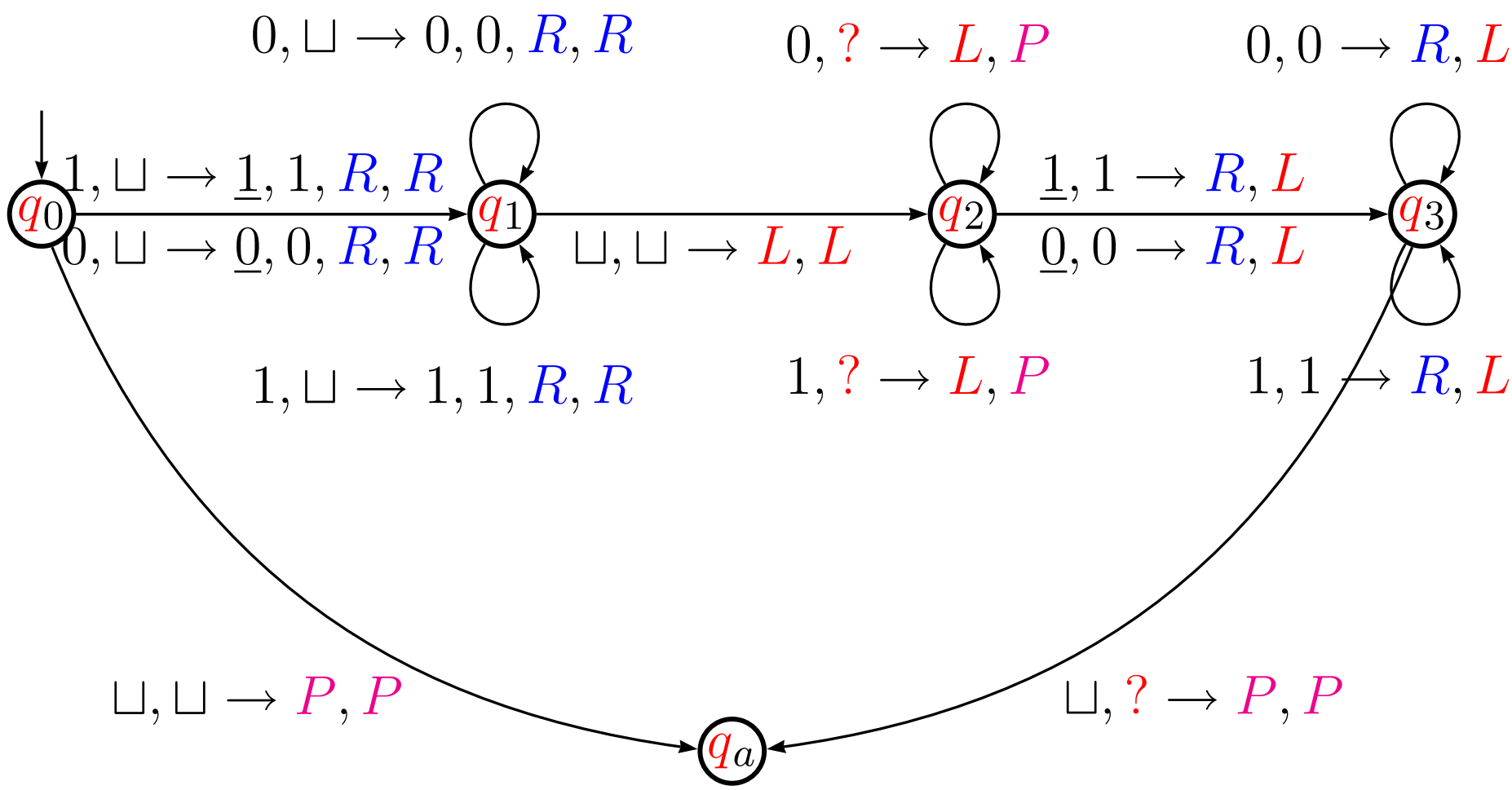
Exemplo de MT multifica

Descrição alto nível de uma máquina M_5 com **duas** fitas que decide

M_5 = "Com entrada w:

1. copie entrada para $2^{\underline{a}}$ fita.
2. mova a cabeça da $1^{\underline{a}}$ fita para a primeira posição, mova a cabeça da $2^{\underline{a}}$ fita para a última posição
3. Mova simultaneamente a cabeça da $1^{\underline{a}}$ fita para a direita e da $2^{\underline{a}}$ fita para a esquerda. Se em algum passo os símbolos sobre as cabeças forem diferentes, então rejeite, senão aceite.

Diagrama de estados para M_5



Transições ausentes levam para $q_{\text{rejeição}}$

Número de passos

Se a cadeia de entrada tem comprimento n então a máquina M_5 faz não mais do que $3n$ passos.

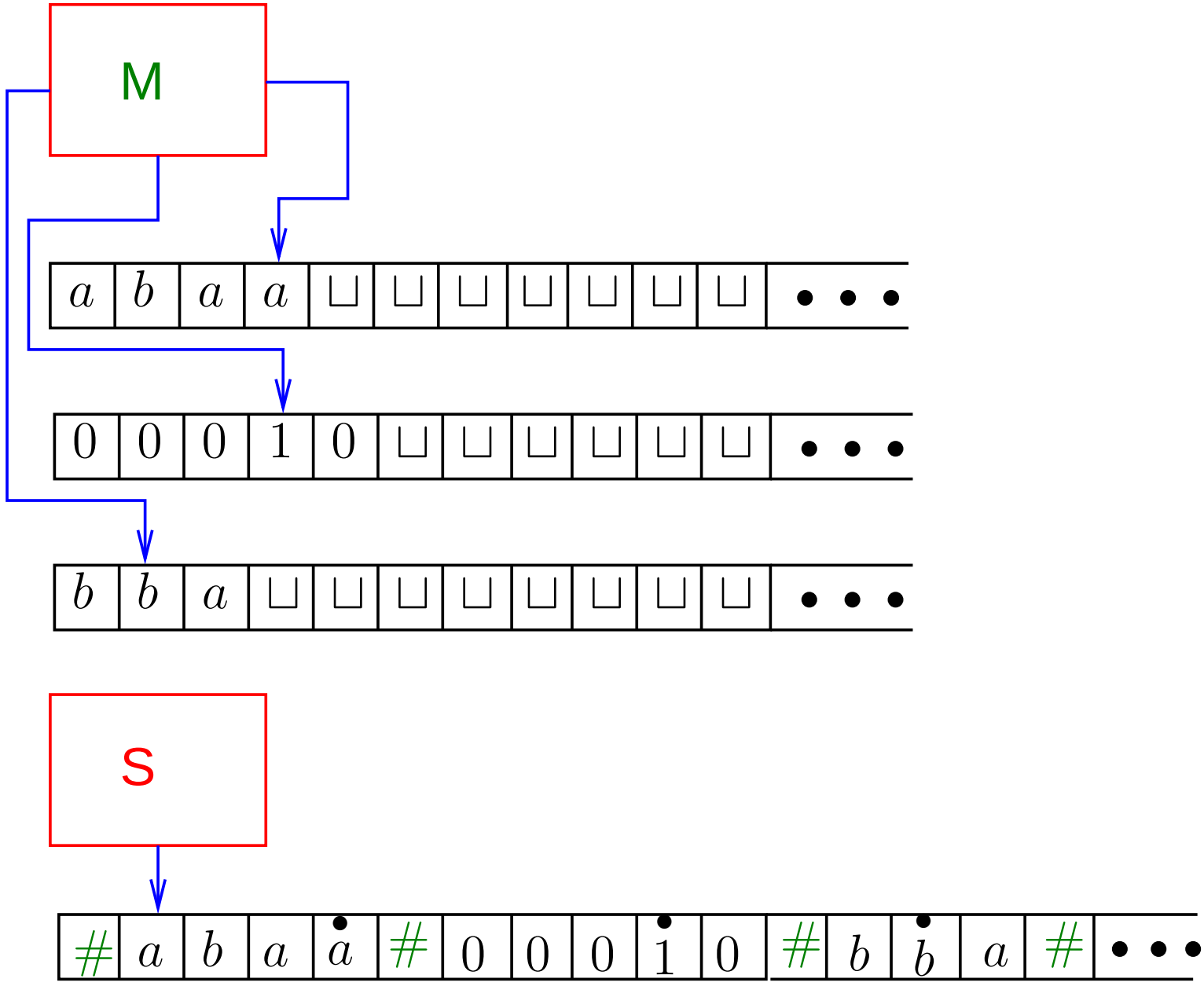
MT multifita por MT fita única

Duas máquinas são **equivalentes** se elas reconhecem a mesma linguagem.

Teorema. Dada uma máquina de Turing multifita M, podemos construir uma uma máquina de Turing S com uma única fita e equivalente a M.

Prova: Vamos mostrar como converter M para S.
Digamos que M tenha k fitas.

Ilustração da idéia da prova



$S =$ "Com entrada $w = w_1 w_2 \cdots w_n$:

1. Acerte a fita de S para representar todas as k fitas de M:

$$\# \dot{w}_1 \; w_2 \cdots w_n \# \dot{\sqcup} \# \dot{\sqcup} \# \cdots \#$$

2. Para simular um movimento de M, S percorre sua fita do primeiro $\#$, que marca o início da fita, até o $k+1$-ésimo $\#$, que marca o fim da fita, para determinar os símbolos sob as cabeças. Então S faz uma segunda passada para atualizar as fitas, de acordo com o determinado pela transição de M.
   S precisa ter novos estados correspondentes aos possíveis estados de M e aos possíveis conteúdos das fitas:

$$Q \times \Gamma^k$$

3. Se em algum ponto S move uma das cabeças à direita para um $\#$, S precisa escrever um branco, deslocando antes o $\#$ e o restante da fita uma posição à direita.”

Corolário. Uma linguagem é Turing-reconhecível se e somente se alguma máquina de Turing multifita a reconhece.

Prova:

($\Rightarrow$) Uma máquina de Turing comum é um caso particular de uma máquina de Turing multifita.

($\Leftarrow$) Segue do teorema anterior.

Notação assintótica

CLRS 3.1

Funções de n

$5n^2 - 9n$ $\quad$ $4n + 8$ $\quad$ $\lfloor n/3 \rfloor + 4$ $\quad$ $2\sqrt{n} + 7$

2^{n-3} $\quad$ $\lg n \quad (= \log_2 n)$

Qual é maior: $n^2 - 9$ ou $4n + 8$?

Funções de n

$5n^2 - 9n$ $\quad$ $4n + 8$ $\quad$ $\lfloor n/3 \rfloor + 4$ $\quad$ $2\sqrt{n} + 7$

2^{n-3} $\quad$ $\lg n \quad (= \log_2 n)$

Qual é maior: $n^2 - 9$ ou $4n + 8$?

Depende do valor de n!

Funções de n

$5n^2 - 9n$ $\quad 4n + 8$ $\quad \lfloor n/3 \rfloor + 4$ $\quad 2\sqrt{n} + 7$

2^{n-3} $\quad \lg n \quad (= \log_2 n)$

Qual é maior: $n^2 - 9$ ou $4n + 8$?

Depende do valor de n!

Qual cresce mais?

Funções de n

$5n^2 - 9n$ $\quad 4n + 8$ $\quad \lfloor n/3 \rfloor + 4$ $\quad 2\sqrt{n} + 7$

2^{n-3} $\quad \lg n \quad (= \log_2 n)$

Qual é maior: $n^2 - 9$ ou $4n + 8$?

Depende do valor de n!

Qual cresce mais?

Comparação **assintótica**, ou seja, para n ENORME.

Funções de n

$5n^2 - 9n$ $\quad 4n + 8$ $\quad \lfloor n/3 \rfloor + 4$ $\quad 2\sqrt{n} + 7$

2^{n-3} $\quad \lg n \quad (= \log_2 n)$

Qual é maior: $n^2 - 9$ ou $4n + 8$?

Depende do valor de n!

Qual cresce mais?

Comparação **assintótica**, ou seja, para n ENORME.

$\lg n$ $\quad 2\sqrt{n} + 7$ $\quad \lfloor n/3 \rfloor + 4$ $\quad 4n + 8$ $\quad 5n^2 - 9n$

2^{n-3}

Notação O

Intuitivamente...

$\mathrm{O}(f(n))$ $\approx$ funções que não crescem mais rápido que $f(n)$

$\approx$ funções menores ou iguais a um múltiplo de $f(n)$

n^2 $\quad$ $(3/2)n^2$ $\quad$ $9999n^2$ $\quad$ $n^2/1000$ $\quad$ etc.

crescem todas com a mesma velocidade

Notação O

Intuitivamente...

$\mathrm{O}(f(n))$ $\approx$ funções que não crescem mais rápido que $f(n)$

$\approx$ funções menores ou iguais a um múltiplo de $f(n)$

n^2 $(3/2)n^2$ $9999n^2$ $n^2/1000$ etc.

crescem todas com a mesma velocidade

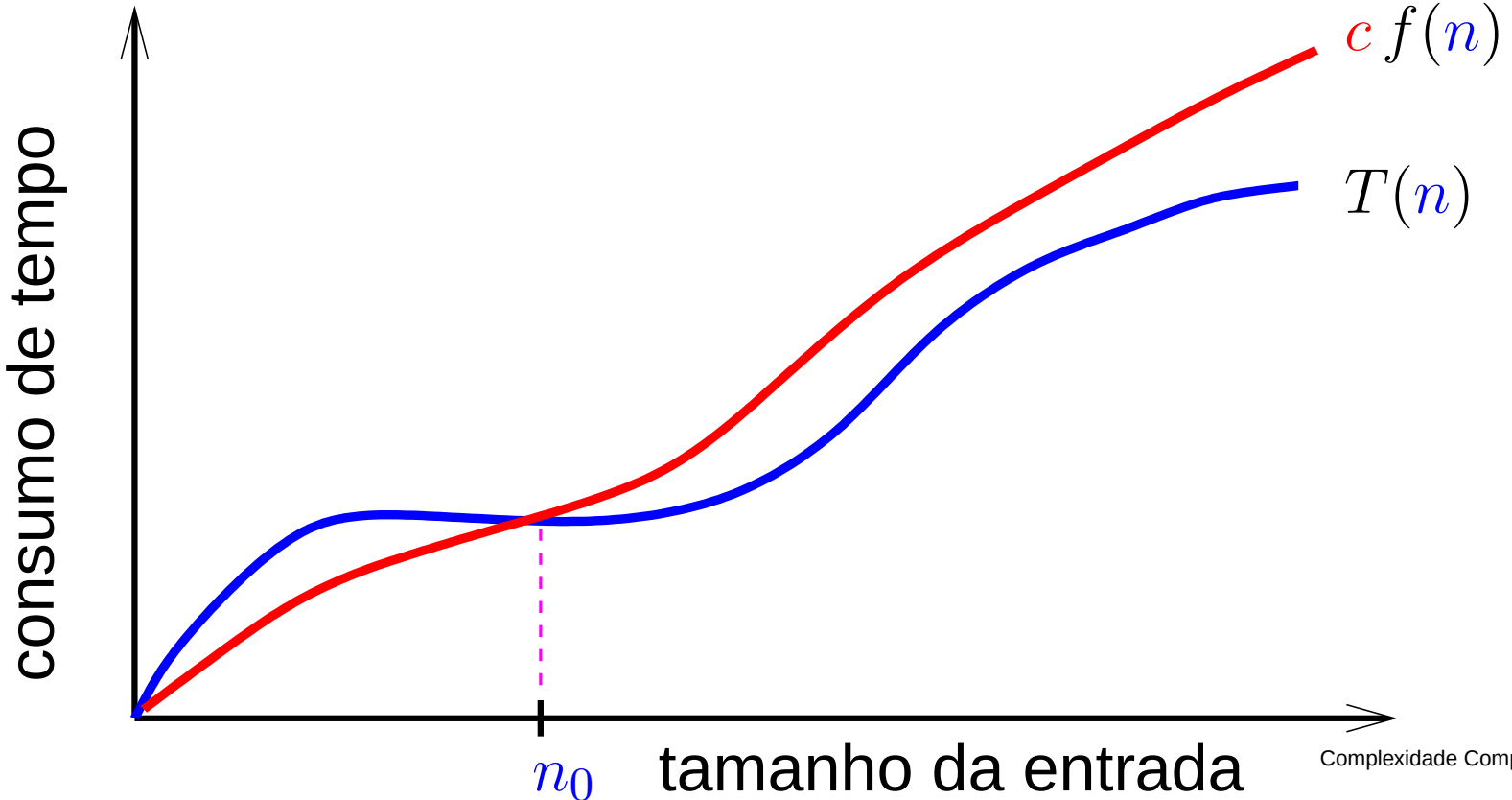
- $n^2 + 99n$ é $\mathrm{O}(n^2)$
- $33n^2$ é $\mathrm{O}(n^2)$
- $9n + 2$ é $\mathrm{O}(n^2)$
- $0{,}00001n^3 - 200n^2$ não é $\mathrm{O}(n^2)$

Definição

Sejam $T(n)$ e $f(n)$ funções dos inteiros nos reais positivos. Dizemos que $T(n)$ **é** $\mathrm{O}(f(n))$ se existem constantes positivas c e n_0 tais que

$$T(n) \leq c\, f(n)$$

para todo $n \geq n_0$.

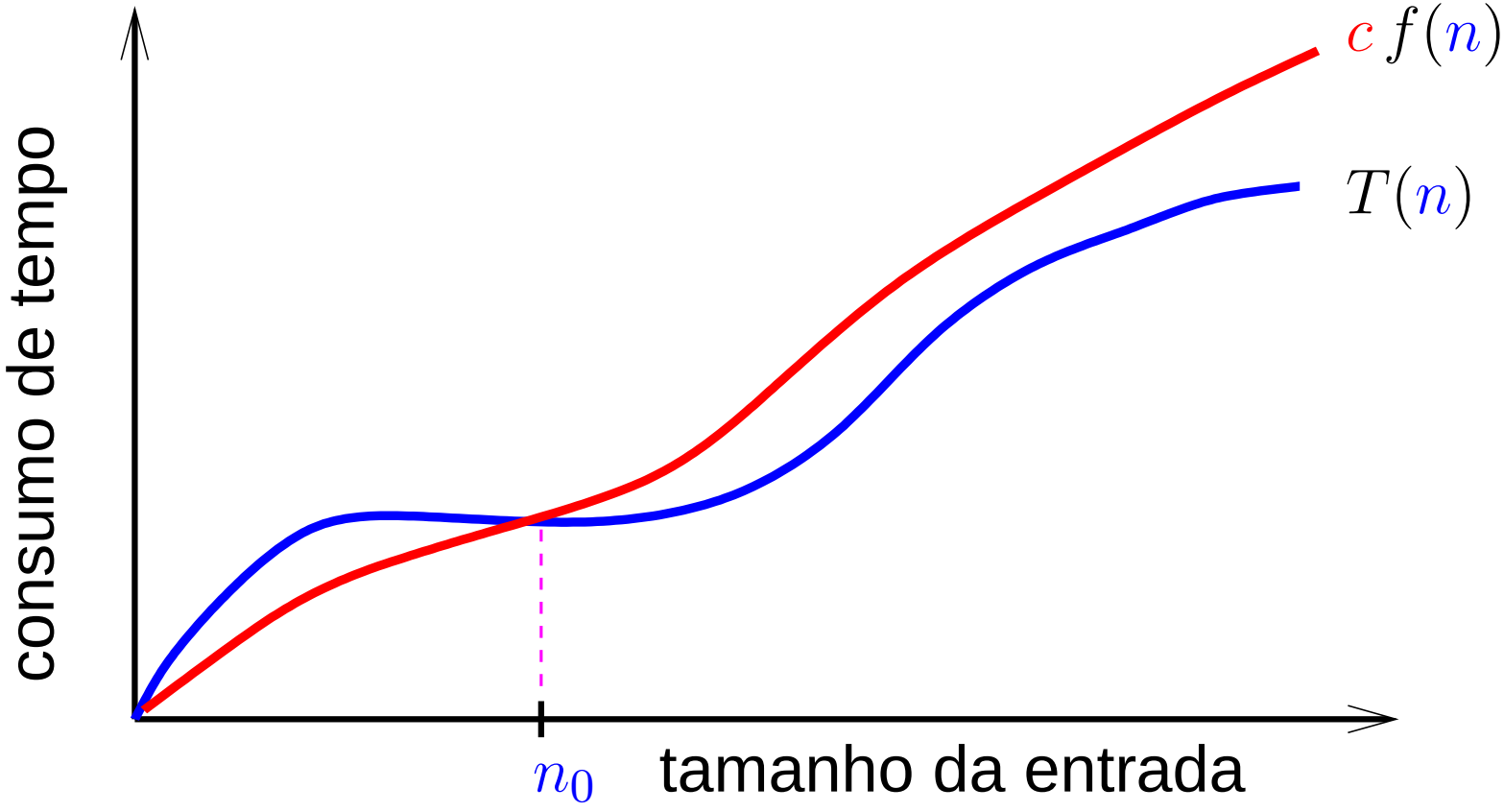


Mais informal

$T(n)$ é $\mathrm{O}(f(n))$ se existe $c > 0$ tal que

$$T(n) \leq c\, f(n)$$

para todo n suficientemente GRANDE.



Exemplos

$T(n)$ é $\mathrm{O}(f(n))$ lê-se "$T(n)$ é O de $f(n)$" ou
"$T(n)$ é da ordem de $f(n)$"

Exemplos

$T(n)$ é $\mathrm{O}(f(n))$ lê-se "$T(n)$ é O de $f(n)$" ou
"$T(n)$ é da ordem de $f(n)$"

Exemplo 1

Se $T(n) \leq 500 f(n)$ para todo $n \geq 10$, então $T(n)$ é $\mathrm{O}(f(n))$.

Exemplos

$T(n)$ é $\mathrm{O}(f(n))$ lê-se "$T(n)$ é O de $f(n)$" ou
"$T(n)$ é da ordem de $f(n)$"

Exemplo 1

Se $T(n) \leq 500 f(n)$ para todo $n \geq 10$, então $T(n)$ é $\mathrm{O}(f(n))$.

Prova: Aplique a definição com $c = 500$ e $n_0 = 10$.

Exemplos

$T(n)$ é $\mathrm{O}(f(n))$ lê-se "$T(n)$ é O de $f(n)$" ou
"$T(n)$ é da ordem de $f(n)$"

Exemplo 1

Se $T(n) \leq 500 f(n)$ para todo $n \geq 10$, então $T(n)$ é $\mathrm{O}(f(n))$.

Prova: Aplique a definição com $c = 500$ e $n_0 = 10$.

Exemplo 2

$10n^2$ é $\mathrm{O}(n^3)$.

Exemplos

$T(n)$ é $\mathrm{O}(f(n))$ lê-se "$T(n)$ é O de $f(n)$" ou
"$T(n)$ é da ordem de $f(n)$"

Exemplo 1

Se $T(n) \leq 500 f(n)$ para todo $n \geq 10$, então $T(n)$ é $\mathrm{O}(f(n))$.

Prova: Aplique a definição com $c = 500$ e $n_0 = 10$.

Exemplo 2

$10n^2$ é $\mathrm{O}(n^3)$.

Prova: Para $n \geq 0$, temos que $0 \leq 10n^2 \leq 10\, n^3$.

Outra prova: Para $n \geq 10$, temos $0 \leq 10n^2 \leq n \times n^2 = 1\, n^3$.

Mais exemplos

Exemplo 3

$\lg n$ é $\mathrm{O}(n)$.

Mais exemplos

Exemplo 3

$\lg n$ é $\mathrm{O}(n)$.

Prova: Para $n \geq 1$, tem-se que $\lg n \leq 1\, n$.

Mais exemplos

Exemplo 3

$\lg n$ é $\mathrm{O}(n)$.

Prova: Para $n \geq 1$, tem-se que $\lg n \leq 1\, n$.

Exemplo 4

$20n^3 + 10n \log n + 5$ é $\mathrm{O}(n^3)$.

Mais exemplos

Exemplo 3

$\lg n$ é $\mathrm{O}(n)$.

Prova: Para $n \geq 1$, tem-se que $\lg n \leq 1\, n$.

Exemplo 4

$20n^3 + 10n \log n + 5$ é $\mathrm{O}(n^3)$.

Prova: Para $n \geq 1$, tem-se que

$$20n^3 + 10n \lg n + 5 \leq 20n^3 + 10n^3 + 5n^3 = 35\, n^3.$$

Outra prova: Para $n \geq 10$, tem-se que

$$20n^3 + 10n \lg n + 5 \leq 20n^3 + n\, n \lg n + n \leq 20n^3 + n^3 + n^3 = 22\, n^3.$$

Mais exemplos ainda

Exemplo 5

$3 \lg n + \lg \lg n$ é $\mathrm{O}(\lg n)$.

Mais exemplos ainda

Exemplo 5

$3 \lg n + \lg \lg n$ é $\mathrm{O}(\lg n)$.

Prova: Para $n \geq 2$, tem-se que

$$3 \lg n + \lg \lg n \leq 3 \lg n + \lg n = 4 \lg n.$$

[Note que $\lg \lg n$ não é definida para $n = 1$.]

Mais exemplos ainda

Exemplo 5

$3 \lg n + \lg \lg n$ é $\mathrm{O}(\lg n)$.

Prova: Para $n \geq 2$, tem-se que

$$3 \lg n + \lg \lg n \leq 3 \lg n + \lg n = 4 \lg n.$$

[Note que $\lg \lg n$ não é definida para $n = 1$.]

Exemplo 6

10^{100} é $\mathrm{O}(1)$.

Mais exemplos ainda

Exemplo 5

$3 \lg n + \lg \lg n$ é $\mathrm{O}(\lg n)$.

Prova: Para $n \geq 2$, tem-se que

$$3 \lg n + \lg \lg n \leq 3 \lg n + \lg n = 4 \lg n.$$

[Note que $\lg \lg n$ não é definida para $n = 1$.]

Exemplo 6

10^{100} é $\mathrm{O}(1)$.

Prova: Para $n \geq 1$, tem-se que

$$10^{100} = 10^{100}\, n^0 = 10^{100} \times 1.$$

[Note que n não precisa aparecer, já que estamos lidando com funções constantes.]

Uso da notação O

$\mathrm{O}(f(n)) = \{T(n)$: existem c e n_0 tq $T(n) \leq cf(n), n \geq n_0\}$

"$T(n)$ é $\mathrm{O}(f(n))$" deve ser entendido como "$T(n) \in \mathrm{O}(f(n))$".

"$T(n) = \mathrm{O}(f(n))$" deve ser entendido como "$T(n) \in \mathrm{O}(f(n))$".

"$T(n) \leq \mathrm{O}(f(n))$" é feio.

"$T(n) \geq \mathrm{O}(f(n))$" não faz sentido!

"$T(n)$ é $g(n) + \mathrm{O}(f(n))$" significa que existe constantes positivas c e n_0 tais que

$$T(n) \leq g(n) + c\, f(n)$$

para todo $n \geq n_0$.

Nomes de classes O

classe	nome
$\mathrm{O}(1)$	constante
$\mathrm{O}(\lg n)$	logarítmica
$\mathrm{O}(n)$	linear
$\mathrm{O}(n \lg n)$	$n \log n$
$\mathrm{O}(n^2)$	quadrática
$\mathrm{O}(n^3)$	cúbica
$\mathrm{O}(n^k)$ com $k >= 1$	polinomial
$\mathrm{O}(2^n)$	exponencial
$\mathrm{O}(a^n)$ com $a > 1$	exponencial

Exercícios

Exercício 2.A

Prove que $n^2 + 10n + 20 = \mathrm{O}(n^2)$

Exercício 2.B

Prove que $300 = \mathrm{O}(1)$

Exercício 2.C

Prove que $\lceil n/3 \rceil = \mathrm{O}(n)$

É verdade que $n = \mathrm{O}(\lfloor n/3 \rfloor)$?

Exercício 2.D

Prove que $\lg n = \mathrm{O}(\log_{10} n)$

Exercício 2.E

Prove que $n = \mathrm{O}(2^n)$

Exercício 2.F

Prove que $\lg n = \mathrm{O}(n)$

Exercício 2.G

Prove que $n/1000$ não é $\mathrm{O}(1)$

Exercício 2.H

Prove que $\frac{1}{2}\, n^2$ não é $\mathrm{O}(n)$

Mais exercícios

Exercício 2.I

Suponha T definida para $n = 0, 1, \ldots$

Se $T(n) = \mathrm{O}(1)$, mostre que existe c' tal que $T(n) \leq c'$ para todo $n \geq 0$.

Se $T(n) = \mathrm{O}(n)$, mostre que existe c' tal que $T(n) \leq c'n$ para todo $n \geq 1$.

Exercício 2.J

Prove que $n^2 + 999n + 9999 = \mathrm{O}(n^2)$.

Exercício 2.K

Prove que $\frac{1}{2}n(n+1) \;=\; \mathrm{O}(n^2)$.

Exercício 2.L

É verdade que $\frac{1}{100}n^2 - 999n - 9999 = \mathrm{O}(n)$? Justifique.

Exercício 2.M

Suponha que $f(n) = n^2$ quando n é par e $f(n) = n^3$ quando n é ímpar.

É verdade que $f(n) = \mathrm{O}(n^2)$?

É verdade que $f(n) = \mathrm{O}(n^3)$?

É verdade que $n^2 = \mathrm{O}(f(n))$?

É verdade que $n^3 = \mathrm{O}(f(n))$?

Mais exercícios ainda

Exercício 2.N
É verdade que $n^2 = \mathrm{O}(2^n)$?

Exercício 2.O
É verdade que $\lg n = \mathrm{O}(\sqrt{n})$?

Exercício 2.P
Suponha $f(n) = 64n \lg n$ e $g(n) = 8n^2$, com n inteiro positivo.
Para que valores de n temos $f(n) \leq g(n)$?

Exercício 2.Q (bom!)
Suponha T e f definidas para $n = 1, 2, \ldots$ Mostre que se $T(n) = \mathrm{O}(f(n))$ e $f(n) > 0$ para $n \geq 1$ então existe c' tal que $T(n) \leq c' f(n)$ para todo $n \geq 1$.

Exercício 2.R (bom!)
Faz sentido dizer "$T(n) = \mathrm{O}(n^2)$ para $n \geq 3$"?

Mais exercícios ainda ainda

Exercício 2.S

É verdade que $2^n = \mathrm{O}(n)$?

É verdade que $n = \mathrm{O}(\lg n)$?

Justifique.

Exercício 2.T

É verdade que $n + \sqrt{n}$ é $\mathrm{O}(n)$?

É verdade que n é $\mathrm{O}(\sqrt{n})$?

É verdade que $n^{2/3}$ é $\mathrm{O}(\sqrt{n})$?

É verdade que $\sqrt{n} + 1000$ é $\mathrm{O}(n)$?

Exercício 2.U

É verdade que $\lg n = \mathrm{O}(n^{1/2})$?

É verdade que $\sqrt{n} = \mathrm{O}(\lg n)$?

É verdade que $\lg n = \mathrm{O}(n^{1/3})$?

Justifique. (Sugestão: prove, por indução, que $\lg \mathtt{x} \leq \mathtt{x}$ para todo número real $\mathtt{x} \geq 1$.)

Exercício 2.V

É verdade que $\lceil \lg n \rceil = \mathrm{O}(\lg n)$?

MT multifita

MS 3.2, MS 7.1

MT multifita por MT fita única

Duas máquinas são **equivalentes** se elas reconhecem a mesma linguagem.

Teorema. Dada uma máquina de Turing multifita M, podemos construir uma uma máquina de Turing S com uma única fita e equivalente a M. Além disso, se tendo como entrada uma cadeia de comprimento n a máquina M faz não mais do que $t(n) \geq n$ passos, então S faz $\mathrm{O}(t^2(n))$ passos.

Prova: Digamos que M tenha k fitas. Já vimos como converter M para S.

Suponha

$$M = (Q, \Sigma, \Gamma, \delta, q_1, q_{\text{aceitação}}, q_{\text{rejeição}})$$

$$S = (Q', \Sigma', \Gamma', \delta', q_1', q_{\text{aceitação}}', q_{\text{rejeição}}')$$

$$\Gamma' = \Gamma \cup \dot{\Gamma}$$

$$\dot{\Gamma} = \{\dot{\gamma} : \gamma \in \Gamma\}$$

Se M faz não mais do que $t(n)$ passos, então o comprimento de cada cadeia em uma fita de M é $\leq t(n)$.

Logo, o maior comprimento da cadeia na fita de S é $k + 1 + kt(n) = k(t(n) + 1) + 1$.

Inicialização

$S =$ Com entrada $w = w_1 w_2 \cdots w_n$ Acerte a fita de S para representar todas as k fitas de M:

$$\# \overset{\bullet}{w_1}\ w_2 \cdots w_n \# \overset{\bullet}{\sqcup} \# \overset{\bullet}{\sqcup} \# \cdots \#$$

O número de passos para inicialização é igual ao número de passos para

- inserir o primeiro $\#$ e deslocar w e outro $\#$;
- inserir $k-1$ cópias de "$\overset{\bullet}{\sqcup}\ \#$";
- voltar a cabeça da fita para primeira posição.

Logo, esse número de passos é $= (2(1 + n + 1 + 2(k - 1)))$.

Esquerda para a direita

A cadeia de S é varrida da esquerda para a direita guardando

- o estado presente da máquina M; e
- os k símbolos que estão sob as cabeças de M.

Para isto podemos ter estados

$$Q \times \Gamma^i \quad \text{com} \quad i = 0, 1, \ldots, k.$$

O número de passos necessários para isto é

$$\leq k(t(n) + 1) + 1.$$

Configurações de S

$$\{q\}\#\ldots\dot{\gamma}_1\ldots\#\ldots\dot{\gamma}_2\ldots\#\ldots\#\ldots\dot{\gamma}_k\ldots\#$$

$$\#\ldots\dot{\gamma}_1\,\{q,\gamma_1\}\ldots\#\ldots\dot{\gamma}_2\ldots\#\ldots\#\ldots\dot{\gamma}_k\ldots\#$$

$$\#\ldots\dot{\gamma}_1\ldots\#\ldots\dot{\gamma}_2\,\{q,\gamma_1,\gamma_2\}\ldots\#\ldots\#\ldots\dot{\gamma}_k\ldots\#$$

$$\#\ldots\dot{\gamma}_1\ldots\#\ldots\dot{\gamma}_2\ldots\#\ldots\#\ldots\dot{\gamma}_k\,\{q,\gamma_1,\ldots,\gamma_k\}\#$$

$$\#\ldots\dot{\gamma}_1\ldots\#\ldots\dot{\gamma}_2\ldots\#\ldots\#\ldots\dot{\gamma}_k\ldots\#\{q,\gamma_1,\ldots,\gamma_k\}$$

Direita para a esquerda

Baseada no estado $\{q, \gamma_1, \ldots, \gamma_k\}$ a máquina S percorre a cadeia na sua fita da esquerda para a direita parando em cada símbolo $\dot{\gamma}_i$, realizando as transições da máquina M:

$$\# \ldots \dot{\gamma}_1 \ldots \# \ldots \dot{\gamma}_2 \ldots \# \ldots \# \ldots \sigma \; \dot{\gamma}_i \; \alpha \ldots \# \ldots \# \ldots \dot{\gamma}_k \ldots \#$$

$$\# \ldots \dot{\gamma}_1 \ldots \# \ldots \dot{\gamma}_2 \ldots \# \ldots \# \ldots \dot{\sigma} \; \gamma_i \alpha \ldots \# \ldots \# \ldots \dot{\sigma}_k \ldots \#$$

se

$$\delta(q, \gamma_1, \ldots, \gamma_i, \ldots, \gamma_k) = \delta(p, \ldots, \sigma, \ldots, L, \ldots)$$

Se não houver "complicação" o número de passos necessários para isto é

$$\leq k(t(n) + 1) + 1 + 2k.$$

Complicação

$\dot{\gamma}_i$ pode estar imediatamente antes de um $\#$ e M deve mover a cabeça da i-ésima fita para a direita:

$$\# \ldots \dot{\gamma}_1 \ldots \# \ldots \dot{\gamma}_2 \ldots \# \ldots \# \ldots \sigma \; \dot{\gamma}_i \; \# \ldots \# \ldots \dot{\gamma}_k \ldots \#$$

$$\# \ldots \dot{\gamma}_1 \ldots \# \ldots \dot{\gamma}_2 \ldots \# \ldots \# \ldots \alpha \; \dot{\#} \ldots \# \ldots \dot{\sigma}_k \ldots \#$$

$$\# \ldots \dot{\gamma}_1 \ldots \# \ldots \dot{\gamma}_2 \ldots \# \ldots \# \ldots \alpha \; \dot{\sqcup} \dot{\#} \ldots \# \ldots \dot{\sigma}_k \ldots \#$$

$$\# \ldots \dot{\gamma}_1 \ldots \# \ldots \dot{\gamma}_2 \ldots \# \ldots \# \ldots \alpha \; \dot{\sqcup} \; \# \ldots \# \ldots \dot{\sigma}_k \ldots \#$$

Neste caso precisamos deslocar vários símbolos para a direita. Número de passos necessários para cada fita de M é $\leq 2(k(t(n)+1)+1)$.

Total: $\leq 2k(k(t(n)+1)+1)$

Número de passos de S

A simulação continua até que M pare.

Inicialização: $\leq (2(1 + n + 1 + 2(k - 1))) = \mathrm{O}(n)$

Cada passo de M:

- Esquerda para a direita: $\leq k(t(n) + 1) + 1 = \mathrm{O}(t(n))$
- Direita para a esquerda: $\leq 2k(k(t(n) + 1) + 1) = \mathrm{O}(t(n))$

Número de passos de $M \leq t(n)$

Logo, número de passos de S é $\mathrm{O}(n + t^2(n))$

Como $t(n) \geq n$, a simulação toda pode ser realizada em $\mathrm{O}(t^2(n))$ passos