

## 1.5 - Procedimentos Decisórios

---

- ♦ Um problema decisório é uma sentença (usualmente apresentada como uma pergunta) que pode ser verdadeira ou falsa. A solução de um problema decisório é “sim”, se a sentença for verdadeira, ou “não”, se a sentença for falsa.
- ♦ Exemplo: A máquina de Turing  $M$  pára com entrada  $w$ ?  
Neste problema existem duas variáveis:  $M$  e  $w$ . Uma **instância** de um problema decisório é obtida particularizando-se as variáveis do problema. No exemplo acima, especificando qual é a máquina  $M$  e qual é a entrada  $w$  temos uma instância do problema.
- ♦ Um problema decisório é solúvel (ou decidível) se existe um algoritmo (ou seja, uma máquina de Turing que sempre pára) que fornece uma resposta “sim” ou “não”. Um problema decisório é parcialmente solúvel (ou indecidível) se existe um procedimento (ou seja, uma máquina de Turing) capaz de fornecer a resposta “sim”.
- ♦ Exemplo: “ $M$  pára com entrada  $w$ ?” é um problema decisório parcialmente solúvel. O procedimento que resolve parcialmente o problema é a máquina de Turing universal (MTU):
  - se  $M$  pára com entrada  $w$ , MTU pára e responde “sim”;
  - se  $M$  não para com entrada  $w$ , MTU não para (ou seja, nada responde).
- ♦ O complemento de um problema decisório  $P$  é o problema (representado por  $\sim P$ ) que se obtém quando são invertidos os significados de “sim” e “não”.

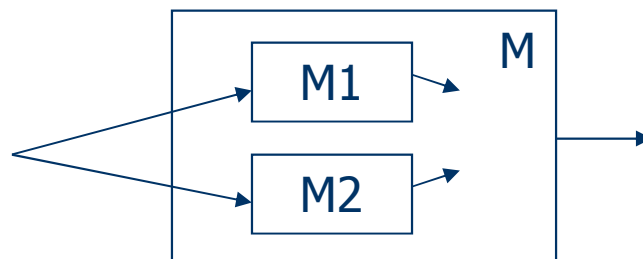
## Procedimentos Decisórios

- ♦ Teorema:  $P$  é solúvel  $\Leftrightarrow P$  e  $\sim P$  são parcialmente solúveis.

- ♦ Prova:

$(\Rightarrow)$   $P$  é solúvel. Então existe uma máquina de Turing que sempre responde sim ou não (ou seja, sempre pára, ou no estado  $q_{\text{sim}}$  ou no estado  $q_{\text{não}}$ ). Podemos então construir uma nova MT trocando o estado  $q_{\text{sim}}$  por  $q_{\text{não}}$  e vice-versa. Portanto, essa nova MT irá sempre responder sim/não para  $\sim P$ . Logo,  $\sim P$  é solúvel.

$(\Leftarrow)$   $P$  e  $\sim P$  são parcialmente solúveis. Sejam  $M1$  e  $M2$  as máquinas de Turing que respondem “sim” para os problemas  $P$  e  $\sim P$ , respectivamente. Podemos então construir uma nova MT  $M$  que simula (não deterministicamente)  $M1$  e  $M2$ , ou seja:



Quando  $M1$  pára (isto é, responde “sim” para  $P$ ),  $M$  pára e responde “sim”. Quando  $M2$  pára (isto é, responde “sim” para  $\sim P$ ),  $M$  pára e responde “não”. Logo,  $M$  irá sempre responder sim/não e, portanto,  $P$  é solúvel.

## Procedimentos Decisórios

---

- ♦ Corolário: O complemento do problema da parada de máquinas de Turing não é parcialmente solúvel (ou seja, não existe um procedimento capaz de responder “sim” ao problema: “M pára com entrada w?”).

### Princípio da redução

- ♦ Sejam A e B dois problemas decisórios. Considere que é possível modificar o algoritmo que resolve o problema A para obter um algoritmo capaz de resolver o problema B. Nesse caso, dizemos que o problema B reduz-se ao problema A, ou seja, uma solução para o problema A implica numa solução para o problema B ( $A \Rightarrow B$ ).
- ♦ Quando um problema B reduz-se a um problema A e sabe-se que B é um problema indecidível, então podemos concluir que o problema A é indecidível também:

$$( (A \Rightarrow B) \wedge (\sim B) ) \Rightarrow (\sim A) \quad \text{“modus tolens”}$$

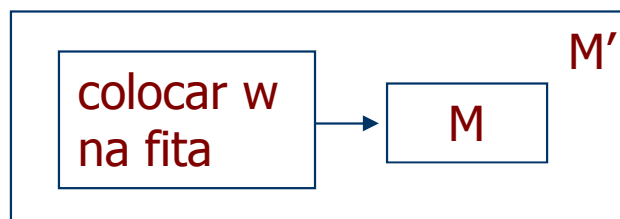
- ♦ Exemplo: “M pára com entrada vazia (isto é, com a fita de entrada totalmente branca)?” é um problema indecidível.

Sejam:                problema A: “M pára com entrada vazia?”

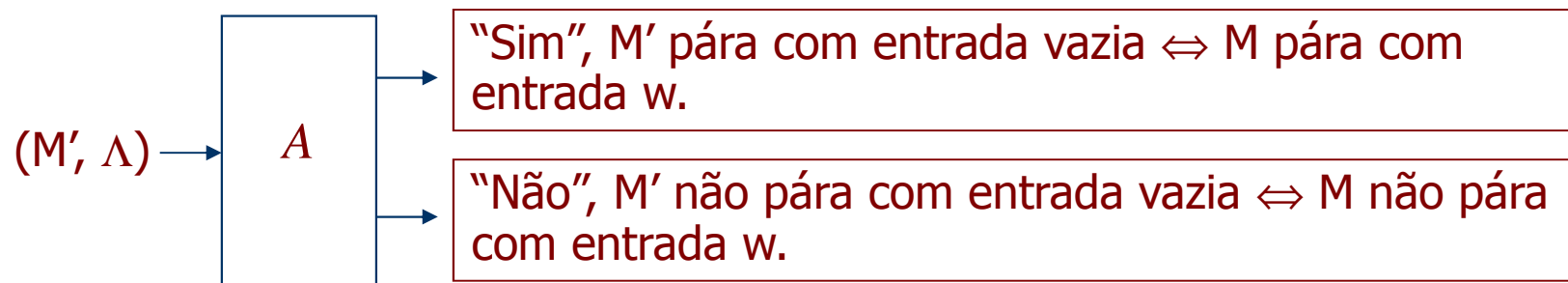
                         problema B: “M pára com entrada w?”

Considere, por absurdo, que o problema A é decidível. Então, existe um algoritmo A que resolve o problema. Podemos então construir uma máquina de Turing M' como:

## Procedimentos Decisórios



Nesse caso, teremos:



Logo, se o algoritmo  $A$  existisse, o problema da parada de máquinas de Turing seria decidível. Como esse problema é sabidamente indecidível,  $A$  não pode existir. Logo, o problema  $A$  é indecidível.

- ♦ Um outro problema importante é o problema da parada uniforme: " $M$  pára com qualquer entrada?". Esse é um problema de grande importância pois corresponde ao problema: "O procedimento  $P$  é um algoritmo?". Esse também é um problema indecidível. Sejam:

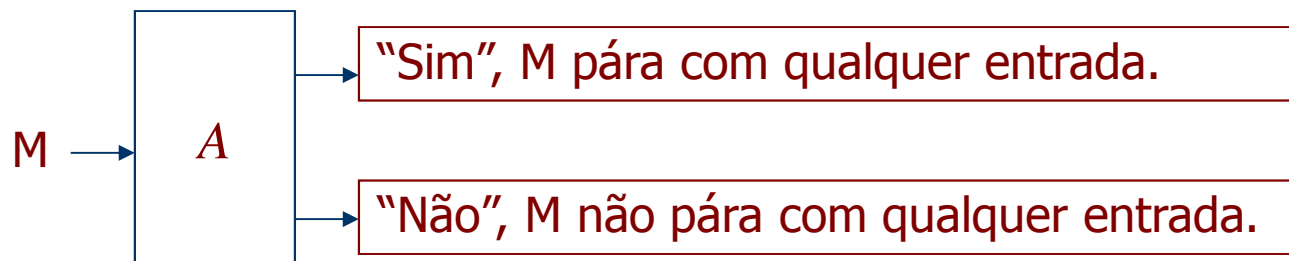
A: " $M$  pára com qualquer entrada?"

B: " $M$  pára com entrada vazia?"

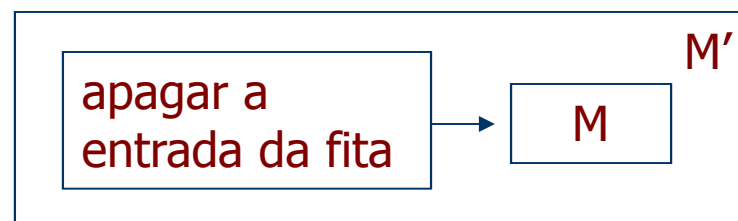
Vamos supor, por absurdo, que existe algoritmo  $A$  tal que:

## Procedimentos Decisórios

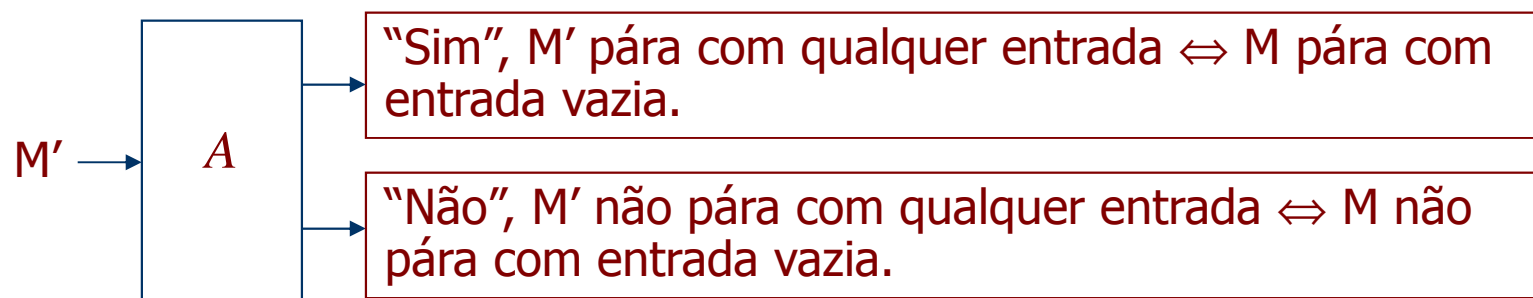
---



Podemos então construir uma máquina  $M'$  tal que:



Então:



Logo,  $A$  resolve o problema  $B$ . Absurdo, pois  $B$  é indecidível. Portanto,  $A$  não existe e  $A$  é indecidível.

## Procedimentos Decisórios

---

- ♦ Muitos outros problemas de importância prática também são indecidíveis. Podemos, por exemplo, para as máquinas de Turing, considerar que, além de poderem ser vistas como reconhecedoras de linguagens, sejam vistas como calculadoras de funções de inteiros em inteiros, da seguinte maneira:
  - o inteiro  $i \geq 0$  pode ser representado pela cadeia  $0^i$
  - se uma função tem  $k$  argumentos  $i_1, i_2, \dots, i_k$ , então a fita de entrada inicialmente pode ser:  $0^{i_1}10^{i_2}1\dots10^{i_k}$
  - se a MT pára (num estado qualquer) com sua fita constituída apenas de  $0^m$ , para algum  $m$ , dizemos que a MT calcula a função  $f(i_1, i_2, \dots, i_k) = m$
  - se  $f(i_1, i_2, \dots, i_k)$  for definida para todos os possíveis  $i_1, i_2, \dots, i_k$  então  $f$  é uma função recursiva total. Uma função  $f(i_1, i_2, \dots, i_k)$  calculada por uma MT é denominada função recursiva parcial pois a MT pode não parar para algumas entradas. As funções recursivas totais são calculadas por máquinas de Turing que sempre páram. Alguns exemplos de funções recursivas totais sobre inteiros:  $n!$ ,  $n_1 \times n_2$ ,  $\log_2 n$ .
- ♦ Um problema prático importante é: “M1 e M2 calculam a mesma função?”.

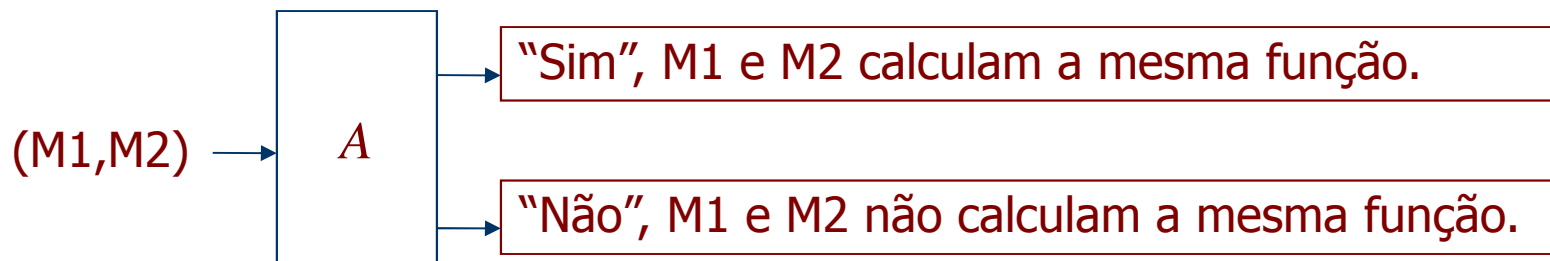
Esse problema também é indecidível. Sejam:

A: “M1 e M2 calculam a mesma função?”

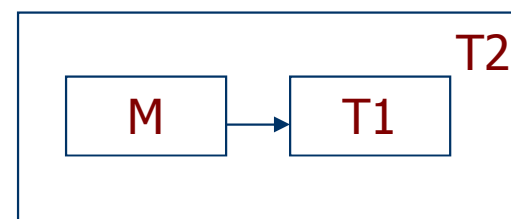
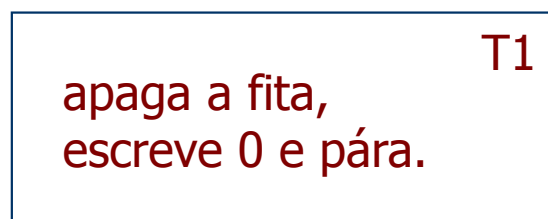
B: “M pára com qualquer entrada?”

Seja  $A$  um algoritmo que resolve o problema A, ou seja:

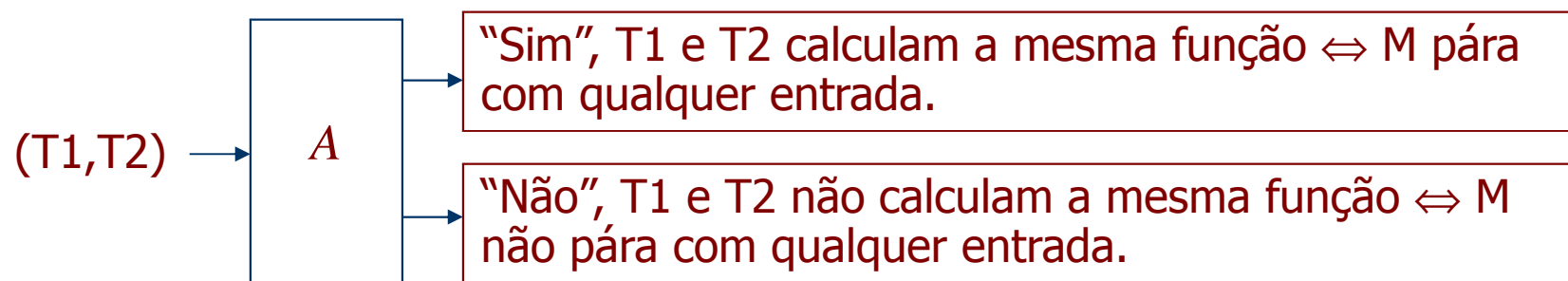
## Procedimentos Decisórios



Podemos então construir as máquinas  $T1$  e  $T2$ :



Então:



Logo, se existisse,  $A$  resolveria o problema da parada uniforme. Absurdo!

# Procedimentos Decisórios

---

## Problema da Correspondência de Post (PCP)

- ♦ Definição: Sejam  $X = (x_1, \dots, x_n)$  e  $Y = (y_1, \dots, y_n)$  duas listas de cadeias não nulas de um mesmo alfabeto  $\Sigma$ . O PCP é determinar se existem ou não índices  $i_1, \dots, i_k$  com  $1 \leq i_j \leq n$  tais que:  $x_{i_1} \dots x_{i_k} = y_{i_1} \dots y_{i_k}$ .
- ♦ Evidentemente, se uma instância do PCP tiver solução, então existirão infinitas soluções pois a sequência de índices pode ser repetida um número qualquer de vezes para obter outras soluções.

- ♦ Exemplos:

$X = (a, bba, aab)$

$Y = (ba, aaa, ba)$

Essa instância, obviamente, não tem solução.

$X = (bbb, abb)$

$Y = (bb, babbb)$

Para essa instância  $(1,2,1)$  é uma solução, pois:

$bbbabbbbb =$

$bbbabbbbb$

$X = (ba, abb, bab)$

$Y = (bab, bb, abb)$

Essa instância não tem solução. Exercício!



## Procedimentos Decisórios

- ♦ Definição: Sejam  $X = (x_1, \dots, x_n)$  e  $Y = (y_1, \dots, y_n)$  duas listas de cadeias não nulas sobre  $\Sigma$ . O problema da correspondência de Post modificado (PCPM) é determinar se existem ou não índices  $i_1, \dots, i_k$  com  $1 \leq i_j \leq n$  tais que:

$$x_1 x_{i_1} \dots x_{i_k} = y_1 y_{i_1} \dots y_{i_k}.$$

- ♦ Lema: O PCPM é redutível ao PCP, ou seja, dada uma instância do PCPM podemos efetivamente construir uma instância do PCP que tem solução se e somente se a instância do PCPM tiver solução.
- ♦ Prova: Seja  $X = (x_1, \dots, x_n)$ ,  $Y = (y_1, \dots, y_n)$  a instância do PCPM. Sejam  $\#$  e  $\$$  dois novos símbolos ( $\# \notin \Sigma$ ,  $\$ \notin \Sigma$ ). Podemos definir as funções:

$$h_L, h_R: \Sigma^+ \rightarrow (\Sigma \cup \{ \# \})^+$$

tais que:  $a \in \Sigma, x \in \Sigma^+$ ,

$$h_L(a) = \#a \quad h_L(xa) = h_L(x)h_L(a)$$

$$h_R(a) = a\# \quad h_R(xa) = h_R(x)h_R(a)$$

**Exemplo:**

$$h_L(abc) = h_L(ab)h_L(c) = h_L(a)h_L(b)h_L(c) = \#a\#b\#c$$

$$h_R(abc) = h_R(ab)h_R(c) = h_R(a)h_R(b)h_R(c) = a\#b\#c\#$$

Essas funções têm as seguintes propriedades:

$$a) \quad h_L(xy) = h_L(x)h_L(y); \quad h_R(xy) = h_R(x)h_R(y) \quad \forall x, y \in \Sigma^+$$

$$b) \quad \#h_R(x) = h_L(x)\#$$

$$c) \quad x = y \Leftrightarrow h_L(x) = h_L(y) \Leftrightarrow h_R(x) = h_R(y)$$

## Procedimentos Decisórios

Construir uma instância do PCP que possui solução se e somente se a instância original do PCPM tiver solução, da seguinte maneira:

$Z = (z_1, \dots, z_{n+2})$ ,  $W = (w_1, \dots, w_{n+2})$  onde:

$$z_1 = \#h_R(x_1) ; z_{i+1} = h_R(x_i) , i = 1, \dots, n ; z_{n+2} = \$$$

$$w_1 = h_L(y_1) ; w_{i+1} = h_L(y_i) , i = 1, \dots, n ; w_{n+2} = \#\$$$

Devemos provar, então, que:

PCP com listas Z e W tem solução  $\Leftrightarrow$  PCPM com listas X e Y tem solução

( $\Leftarrow$ ) Seja  $1, i_1, \dots, i_k$  solução do PCPM com listas X e Y. Logo:

$x_1 x_{i_1} \dots x_{i_k} = y_1 y_{i_1} \dots y_{i_k}$ . Portanto, pela propriedade (b) temos:

$\#h_R(x_1 x_{i_1} \dots x_{i_k}) = h_L(y_1 y_{i_1} \dots y_{i_k})\#$  e, portanto:

$$\#h_R(x_1 x_{i_1} \dots x_{i_k})\$ = h_L(y_1 y_{i_1} \dots y_{i_k})\#\$$$

ou seja:

$$\underbrace{\#h_R(x_1)}_{z_1} \underbrace{h_R(x_{i_1})}_{z_{i_1+1}} \dots \underbrace{h_R(x_{i_k})}_{z_{i_k+1}} \underbrace{\$}_{z_{n+2}} = \underbrace{h_L(y_1)}_{w_1} \underbrace{h_L(y_{i_1})}_{w_{i_1+1}} \dots \underbrace{h_L(y_{i_k})}_{w_{i_k+1}} \underbrace{\#\$}_{w_{n+2}}$$

Portanto:

$$z_1 z_{i_1+1} \dots z_{i_k+1} z_{n+2} = w_1 w_{i_1+1} \dots w_{i_k+1} w_{n+2}$$

ou seja:  $(1, i_1+1, \dots, i_k+1, n+2)$  é solução do PCP com listas Z e W.

## Procedimentos Decisórios

---

( $\Rightarrow$ ) Seja  $i_1, \dots, i_k$  solução do PCP com listas  $Z$  e  $W$ . Então:

$i_1 = 1$  pois  $(z_1, w_1)$  é o único par de termos que começam com o mesmo símbolo ( $\#$ ).

$i_k = n+2$  pois  $(z_{n+2}, w_{n+2})$  é o único par de termos que terminam com o mesmo símbolo ( $\$$ ).

Seja  $j$  o menor inteiro tal que  $i_j = n+2$ . Temos então:

- a)  $i_1, \dots, i_j$  é solução do PCP com listas  $Z$  e  $W$
- b) não existe  $m$ ,  $1 \leq m \leq j$  tal que  $i_m = n+2$  (pela definição de  $j$ )
- c) não existe  $m$ ,  $2 \leq m \leq j$  tal que  $i_m = 1$

Validade de (a):

Se  $i_1, \dots, i_j$  não for solução, então deve ser uma solução parcial, ou seja:

$z_{i_1} \dots z_{i_j-1} \$$  é prefixo de  $w_{i_1} \dots w_{i_j-1} \# \$$  ou

$w_{i_1} \dots w_{i_j-1} \# \$$  é prefixo de  $z_{i_1} \dots z_{i_j-1} \$$

mas isso não pode acontecer pois do contrário o símbolo  $\$$  deveria aparecer no meio de uma das cadeias, o que é impossível porque as únicas cadeias que contêm o símbolo  $\$$  são  $w_{n+2}$  e  $z_{n+2}$  e  $j$  é o menor inteiro tal que  $i_j = n+2$ .

Validade de (c):

Vamos supor que existe  $m$ ,  $2 \leq m \leq j$  tal que  $i_m = 1$ . Então,  $z_{i_1} \dots z_{i_j}$  pode ser escrito na forma  $\alpha \# \beta$ , onde  $\alpha \# = z_{i_1} \dots z_{i_m-1}$  e  $\# \beta = z_{i_m} \dots z_{i_j}$

## Procedimentos Decisórios

Nesse caso,  $z_{i_1} \dots z_{i_j} \neq w_{i_1} \dots w_{i_j}$  (porque na cadeia  $w_{i_1} \dots w_{i_j}$  é impossível aparecer  $\#\#$ ), o que é um absurdo pois  $i_1, \dots, i_j$  é solução do PCP com listas Z e W.

Então:

$$\underbrace{z_1}_{\#h_R(x_1)} \underbrace{z_{i_2-1}}_{h_R(x_{i_2-1})} \dots \underbrace{z_{i_{j-1}-1}}_{h_R(x_{i_{j-1}-1})} = \underbrace{w_1}_{h_L(y_1)} \underbrace{w_{i_2-1}}_{h_L(y_{i_2-1})} \dots \underbrace{w_{i_{j-1}-1}}_{h_L(y_{i_{j-1}-1})} \#\$$$

$$\Rightarrow \#h_R(x_1) h_R(x_{i_2-1}) \dots h_R(x_{i_{j-1}-1}) = h_L(y_1) h_L(y_{i_2-1}) \dots h_L(y_{i_{j-1}-1}) \#$$

$$\Rightarrow \#h_R(x_1 x_{i_2-1} \dots x_{i_{j-1}-1}) = h_L(y_1 y_{i_2-1} \dots y_{i_{j-1}-1}) \#$$

$$\Rightarrow x_1 x_{i_2-1} \dots x_{i_{j-1}-1} = y_1 y_{i_2-1} \dots y_{i_{j-1}-1}$$

$$\Rightarrow (1, i_2-1, \dots, i_{j-1}-1) \text{ é solução do PCPM com listas X e Y.}$$

- ♦ Lema: Se o PCPM fosse decidível, então o problema da parada de máquinas de Turing seria decidível.
- ♦ Prova: Seja  $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$  uma MT e  $\#$ , um novo símbolo,  $\# \notin Q$ ,  $\# \notin \Sigma$ .

Seja a instância do PCPM:

# Procedimentos Decisórios

	Lista X	Lista Y	
	#	#q <sub>0</sub> w#	w ∈ Σ*
<b>Grupo I</b>	x #	x #	x ∈ Γ - { b }
<b>Grupo II</b>	qx zqx q# zq#	yp pzy yp# pzy#	se δ(q,x) = (p,y,→) se δ(q,x) = (p,y,←) se δ(q,b) = (p,y,→) se δ(q,b) = (p,y,←) com x,y,z ∈ Γ - { b }
<b>Grupo III</b>	xqy xq# #qy q##	q q# #q #	q ∈ F

As cadeias do grupo II são simulações de transições de M, onde as cadeias da lista Y são obtidas a partir das cadeias correspondentes da lista X, em uma transição.

A idéia das cadeias do grupo III é fazer a diferença entre os comprimentos das listas X e Y ir diminuindo passo a passo (até que elas se igualem), acrescentando mais símbolos à lista X do que à lista Y.

## Procedimentos Decisórios

Seja a entrada  $w = x_1 \dots x_m$  e seja  $q_0 x_1 \dots x_m \mapsto \alpha_1 q_1 \beta_1 \mapsto \dots \mapsto \alpha_k q_k \beta_k$ ,  $q_i \notin F$  ( $i = 0, 1, \dots, k-1$ ), a sequência de configurações de  $M$  com entrada  $w$ . Vamos mostrar que enquanto  $M$  não alcança um estado final não é possível conseguir uma solução para o PCPM com listas  $X$  e  $Y$  (pois a lista  $Y$  estará sempre maior que a lista  $X$ ).

Inicialmente (como para o PCPM,  $i_1 = 1$ ) temos:

Lista X:  $\#$

Lista Y:  $\#q_0 x_1 \dots x_m \#$

Seja (sem perda de generalidade)  $\delta(q_0, x_1) = (q_1, y_1, \rightarrow)$ . A única maneira de fazer aparecer  $q_0 x_1$  na lista X é usar uma cadeia do grupo II (em correspondência devemos colocar  $y_1 q_1$  na lista Y). Para fazer aparecer  $x_2 \dots x_m \#$  na lista X devemos usar cadeias do grupo I (que serão repetidas na lista Y).

Após esse passo teremos:

Lista X:  $\#q_0 x_1 \dots x_m \#$

Lista Y:  $\#q_0 x_1 \dots x_m \# y_1 q_1 x_2 \dots x_m \#$

Esse processo irá se repetir enquanto utilizarmos cadeias dos grupos I e II apenas, ou seja, enquanto  $M$  não atinge um estado final a lista Y será sempre maior do que a lista X.

Considere agora que  $q_k \in F$ . Nesse ponto teremos:

Lista X:  $\#q_0 x_1 \dots x_m \# \alpha_1 q_1 \beta_1 \# \dots \# \alpha_{k-1} q_{k-1} \beta_{k-1} \#$

Lista Y:  $\#q_0 x_1 \dots x_m \# \alpha_1 q_1 \beta_1 \# \dots \# \alpha_{k-1} q_{k-1} \beta_{k-1} \# \alpha_k q_k \beta_k \#$

## Procedimentos Decisórios

Seja  $\alpha_k q_k \beta_k = \alpha x q_k y \beta$ . Então:

Lista X:  $\#q_0 x_1 \dots x_m \# \alpha_1 q_1 \beta_1 \# \dots \# \alpha_{k-1} q_{k-1} \beta_{k-1} \#$

Lista Y:  $\#q_0 x_1 \dots x_m \# \alpha_1 q_1 \beta_1 \# \dots \# \alpha_{k-1} q_{k-1} \beta_{k-1} \# \alpha x q_k y \beta \#$

Agora, como  $q_k \in F$ , devemos usar cadeias do grupo III. Devemos colocar  $\alpha x q_k y \beta$  na lista X e  $\alpha q_k \beta$  em Y. Logo:

Lista X:  $\#q_0 \dots \# \alpha x q_k y \beta \#$

Lista Y:  $\#q_0 \dots \# \alpha x q_k y \beta \# \alpha q_k \beta \#$

ou seja, a diferença de tamanho entre as listas vai diminuindo. Logo, se M atinge um estado final, podemos igualar as duas listas usando cadeias do grupo III. Portanto:

M pára com entrada  $w \Leftrightarrow$  o PCPM com listas X e Y tem solução.

- ♦ Teorema: O PCP não é decidível.
- ♦ Prova: Consequência dos dois lemas anteriores.

### Aplicação do PCP

- ♦ Sejam as listas  $X = (x_1, \dots, x_k)$  e  $Y = (y_1, \dots, y_k)$  sobre um alfabeto  $\Sigma$ . Seja  $A = \{a_1, \dots, a_k\}$  tal que  $A \cap \Sigma = \emptyset$ .  
Sejam as GLC,  $G_A = (\{S_A\}, T, P_A, S_A)$  e  $G_B = (\{S_B\}, T, P_B, S_B)$  tais que  $T = A \cup \Sigma$ ,  
 $P_A = \{ S_A \rightarrow x_i S_A a_i \mid x_i a_i \}$  e  $P_B = \{ S_B \rightarrow y_i S_B a_i \mid y_i a_i \}$ ,  $i = 1, \dots, k$ .

## Procedimentos Decisórios

---

Portanto:  $L_A = L(G_A) = \{ x_{i1} \dots x_{im} a_{im} \dots a_{i1} \mid m \geq 1 \}$

$L_B = L(G_B) = \{ y_{i1} \dots y_{im} a_{im} \dots a_{i1} \mid m \geq 1 \}$

- ♦ Teorema: O problema “uma GLC  $G$  é ambígua?” é indecidível.
- ♦ Prova: Seja  $G = (\{S, S_A, S_B\}, T, P, S)$  onde:  
 $P = \{ S \rightarrow S_A \mid S_B, S_A \rightarrow x_i S_A a_i \mid x_i a_i, S_B \rightarrow y_i S_B a_i \mid y_i a_i \}$  com  $i = 1, \dots, k$ .  
Então:  $L(G) = L_A \cup L_B$   
Portanto:  $G$  é ambígua  $\Leftrightarrow$  o PCP com listas  $X$  e  $Y$  tem solução.

- ♦ Uma boa leitura:

Douglas Hofstadter

Gödel, Escher, Bach - Um entrelaçamento de gênios brilhantes

Editora UnB, 2000