

## CAPÍTULO 6 - DECIDIBILIDADE

Um problema de decisão, é uma sentença (usualmente apresentada como uma pergunta) que pode ser verdadeira ou falsa. A solução de um problema de decisão (resposta à pergunta) é "SIM" caso a sentença seja verdadeira e "NÃO" se a sentença for falsa.

EXEMPLO: A máquina de Turing  $M$  pára, com entrada  $\alpha$ ?

Neste problema existem duas variáveis:  $M$  e  $\alpha$ . Uma instância de um problema de decisão é obtida particularizando-se as variáveis do problema. No exemplo acima, especificando qual é a máquina  $M$  e qual é a entrada  $\alpha$  tem-se uma instância do problema.

Um problema de decisão 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 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  $\alpha$ ?" é um problema parcialmente solúvel. O procedimento que resolve o problema é a máquina de Turing universal, MTU:

- se  $M$  pára com entrada  $\alpha$ , MTU pára e responde "SIM"
- se  $M$  não pára com entrada  $\alpha$ , MTU não pára (nada responde)

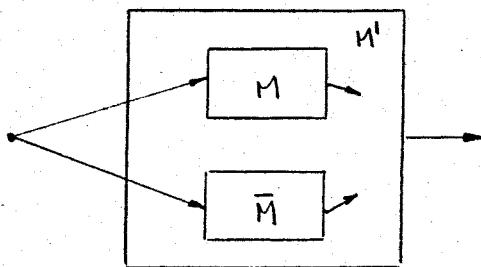
O complemento de um problema de decisão é o problema que se obtém quando são invertidos os significados de "SIM" e "NÃO" (ou seja, SIM/NÃO passa a ser NÃO/SIM)

Teorema 6.1:  $P$  é solúvel  $\Leftrightarrow P$  e  $\bar{P}$  são parcialmente solúveis.

Prova:

( $\Rightarrow$ )  $P$  é solúvel. Então existe uma MT que sempre responde SIM ou NÃO (ou seja, sempre pára, ou no estado  $q_{SIM}$  ou no estado  $q_{NÃO}$ ). Pode-se então construir uma nova MT trocando o estado  $q_{SIM}$  por  $q_{NÃO}$  e vice-versa. Portanto, essa nova MT irá responder sempre SIM/NÃO para  $\bar{P}$ . Logo,  $\bar{P}$  é solúvel.

( $\Leftarrow$ )  $P$  e  $\bar{P}$  são parcialmente solúveis. Sejam  $M$  e  $\bar{M}$  as MT que respondem "SIM" para  $P$  e  $\bar{P}$ , respectivamente. Pode-se então construir uma nova MT,  $M'$ , que simula (não deterministicamente)  $M$  e  $\bar{M}$



Quando  $M$  pára (isto é, responde "SIM"), então  $M'$  pára e responde "SIM". Quando  $\bar{M}$  pára (isto é, responde "SIM"),  $M'$  pára e responde "NÃO". Logo,  $M'$  irá sempre responder SIM/NÃO e portanto  $P$  é solúvel.

Corolário 6.1: 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:  
"  $M$  não pára com entrada  $\alpha$ ? " )

### PRINCÍPIO DA REDUÇÃO

Sejam  $A$  e  $B$  dois problemas de decisão. Considere que é possível modificar o algoritmo que resolve o problema  $A$  para obter um algoritmo capaz de resolver o problema  $B$ . Neste caso diz-se 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 pode-se concluir que o problema  $A$  é indecidível também.

$$( (A \Rightarrow B) \wedge (\neg B) ) \Rightarrow (\neg A) \quad \text{"modus tollens"}$$

EXEMPLO: "  $M$  pára com entrada vazia (isto é, com a fita de entrada totalmente branca) ? "

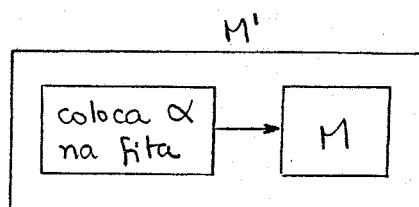
Este problema é indecidível, pois:

sejam :

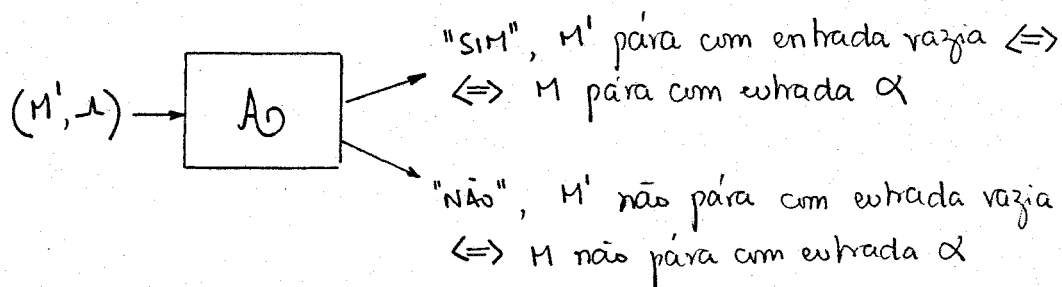
problema A : "M pára com entrada vazia?"

problema B : "M pára com entrada  $\alpha$ ?"

Considere que o problema A é decidível (por absurdo). Então, existe um algoritmo  $A_0$  que resolve o problema. Pode-se então construir uma máquina de Turing  $M'$  como :



Neste caso tem-se :



Logo, se o algoritmo  $A_0$  existisse, o problema da parada de máquinas de Turing seria decidível. Como este problema é sabidamente indecidível,  $A_0$  não pode existir. Logo A é indecidível.

Um outro problema indecidível é o problema da parada uniforme :

"M pára para uma entrada qualquer?"

Ente é um problema de grande importância, pois corresponde ao problema:

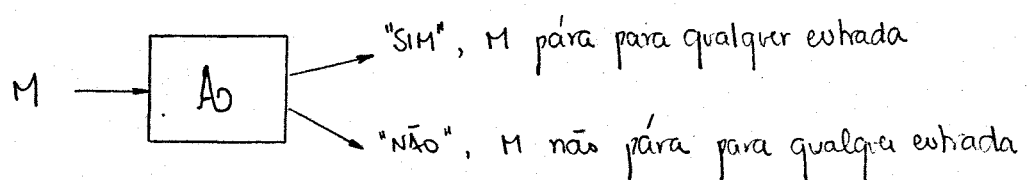
"O procedimento  $P$  é um algoritmo?"

mas, infelizmente, é um problema indecidível. Sejam:

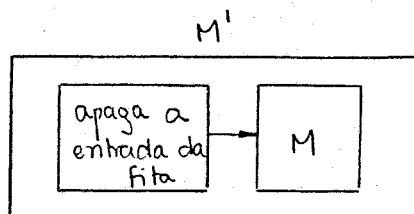
A: "M pára para qualquer entrada?"

B: "M pára com entrada vazia?"

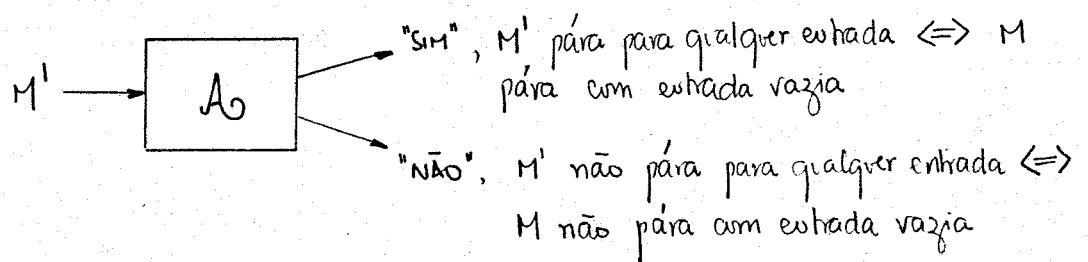
Seja  $A_0$  um algoritmo tal que:



Pode-se então construir uma máquina  $M'$  tal que:



Então:



Logo,  $A_0$  resolve o problema B. Absurdo pois B é indecidível. Portanto  $A_0$  não existe e A é indecidível.

Muitos outros problemas de importância prática são também indecidíveis. Considere, por exemplo:

Alem de um reconhecedor de linguagens, a máquina de Turing pode ser encarada também como um calculador 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:

$$0^{i_1} 1 0^{i_2} 1 \dots 1 0^{i_k}$$

- se a MT pára (num estado qualquer) com sua fita constituída apenas de  $0^m$ , para algum  $m$ , diz-se que a MT calcula a função  $f(i_1, i_2, \dots, i_k) = m$ .
- se  $f(i_1, \dots, i_k)$  for definida para todos os possíveis  $i_1, \dots, i_k$  então  $f$  é uma função recursiva total. Uma função  $f(i_1, \dots, i_k)$  calculada por uma MT é chamada função recursiva parcial, pois a MT pode não parar para algumas entradas (as funções recursivas totais são calculadas por MT que sempre param; algumas funções recursivas totais sobre os inteiros:  $n!$ ,  $n_1 \times n_2$ ,  $\log_2 n$ )

Um problema prático importante é:

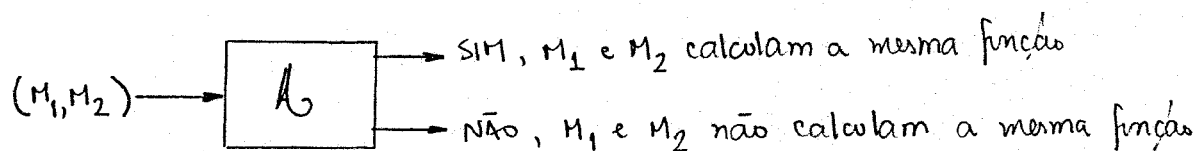
" $M_1$  e  $M_2$  calculam a mesma função?"

Este problema é indecidível. Sejam:

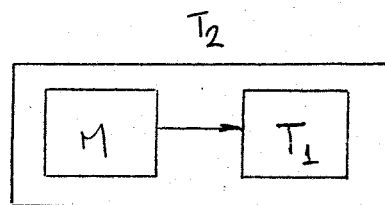
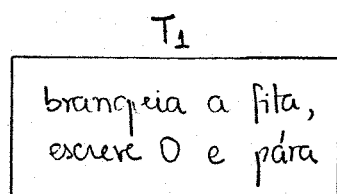
A: " $M_1$  e  $M_2$  calculam a mesma função?"

B: " $M$  pára para todas as entradas!"

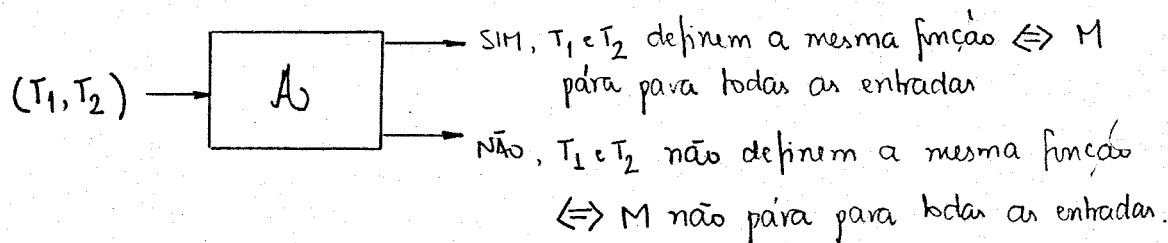
Seja  $A_0$  o algoritmo que resolve o problema A, ou seja:



Pode-se construir as máquinas:



Então:



Portanto, se existisse o algoritmo  $A_0$ , o problema da parada uniforme seria decidível.

## PROBLEMA DA CORRESPONDÊNCIA DE POST (PCP)

definição 6.1 : Sejam  $X = (x_1, \dots, x_n)$  e  $Y = (y_1, \dots, y_n)$  duas listas de cadeias não nulas de um alfabeto comum  $\Sigma$ . O problema da correspondência de Post é determinar se existem ou não, índices  $i_1, \dots, i_k$  com  $1 \leq i_j \leq n$  tal que

$$x_{i_1} \dots x_{i_k} = y_{i_1} \dots y_{i_k}$$

Evidentemente, se uma instância do PCP tiver solução, estão existindo infinitas soluções, pois que a sequência de índices pode ser repetida um número qualquer de vezes para obter outras soluções.

### EXEMPLOS:

$$(1) \quad \begin{aligned} X &= (a, bba, aab) \\ Y &= (ba, aaa, ba) \end{aligned}$$

Essa instância, obviamente, não tem solução.

$$(2) \quad \begin{aligned} X &= (bbb, abb) \\ Y &= (bb, babbb) \end{aligned}$$

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

$$bbb|abb|bbb = bb|babbb|bb$$



$$(3) \quad \begin{aligned} X &= (ba, abb, bab) \\ Y &= (bab, bb, abb) \end{aligned}$$

Essa instância não tem solução. Exercício!

definição 6.2: 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 existe ou não, índices  $i_1, \dots, i_k$ ,  $1 \leq i_j \leq n$  tal que:

$$x_{i_1} x_{i_2} \dots x_{i_k} = y_{i_1} y_{i_2} \dots y_{i_k}$$

Lema 6.1 O PCPM é redutível ao PCP, ou seja, dada uma instância do PCPM pode-se efetivamente construir uma instância do PCP que tem solução se e somente se a instância do PCPM tem solução.

Prova:

Seja  $X = (x_1, \dots, x_n)$ ,  $Y = (y_1, \dots, y_n)$  a instância do PCPM. Sejam  $\phi$  e  $\psi$ , dois novos símbolos ( $\phi \notin \Sigma$ ,  $\psi \notin \Sigma$ )

Pode-se definir as funções:

$$h_L: \Sigma^+ \rightarrow (\Sigma \cup \{\phi, \psi\})^+$$

$$h_R: \Sigma^+ \rightarrow (\Sigma \cup \{\phi, \psi\})^+$$

tais que:

$a \in \Sigma$ ,  $x \in \Sigma^+$  tem-se:

$$h_L(a) = \zeta a \quad ; \quad h_L(xa) = h_L(x) h_L(a)$$

$$h_R(a) = a \zeta \quad ; \quad h_R(xa) = h_R(x) h_R(a)$$

Exemplo:

$$\begin{aligned} h_L(abc) &= h_L(ab) h_L(c) = h_L(a) h_L(b) h_L(c) = \\ &= \zeta a \zeta b \zeta c \end{aligned}$$

$$\begin{aligned} h_R(abc) &= h_R(ab) h_R(c) = h_R(a) h_R(b) h_R(c) = \\ &= a \zeta b \zeta c \zeta \end{aligned}$$

Essas funções tem as seguintes propriedades:

$$\begin{aligned} (a) \quad h_L(xy) &= h_L(x) h_L(y) \\ h_R(xy) &= h_R(x) h_R(y) \end{aligned} \quad \forall x, y \in \Sigma^+$$

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

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

Construir uma instância do PCP que possui solução  $\Leftrightarrow$  a instância original do PCPM possui solução, da seguinte maneira:

$$Z = (z_1, \dots, z_{n+2})$$

$$W = (w_1, \dots, w_{n+2})$$

onde :

$$z_1 = \$ h_R(x_1) \quad ; \quad z_{i+1} = h_R(x_i) \quad (1 \leq i \leq n) \quad ; \quad z_{n+2} = \$$$

$$w_1 = h_L(y_1) \quad ; \quad w_{i+1} = h_L(y_i) \quad (1 \leq i \leq n) \quad ; \quad w_{n+2} = \$$$

Deve-se 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  $Y$

$$\text{Logo:} \quad x_1 x_{i_1} \dots x_{i_k} = y_1 y_{i_1} \dots y_{i_k}$$

Portanto, pela propriedade (b) tem-se

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

e então

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

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 :

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

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

( $\Rightarrow$ ) Seja  $i_1, i_2, \dots, i_k$  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 cujos termos terminam com o mesmo símbolo ( $\$$ )

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

(1)  $i_1, i_2, \dots, i_j$  é solução do PCP com listas  $Z$  e  $W$

(2)  $\nexists l, 1 \leq l \leq j$  tal que  $i_l = n+2$  (pela definição de  $j$ )

(3)  $\nexists l, 2 \leq l \leq j$  tal que  $i_l = 1$

VALIDADE DE (1) :

se  $i_1, i_2, \dots, i_j$  não for solução, então deve ser uma solução parcial, ou seja:

$$z_{i_1} z_{i_2} \dots z_{i_{j-1}} \$ \text{ é prefixo de } w_{i_1} w_{i_2} \dots w_{i_{j-1}} \$$$

ou  $w_{i_1} w_{i_2} \dots w_{i_{j-1}} \zeta \zeta$  é prefixo de  $z_{i_1} z_{i_2} \dots z_{i_{j-1}} \zeta$

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

VALIDADE DE (3)

Supor que existe  $l$ ,  $2 \leq l \leq j$  tal que  $i_l = 1$ . Então

$z_{i_1} z_{i_2} \dots z_{i_j}$  pode ser escrito na forma  $\alpha \zeta \beta$

onde  $\alpha \zeta = z_{i_1} z_{i_2} \dots z_{i_{l-1}}$  e  $\zeta \beta = z_{i_l} \dots z_{i_j}$ . Mas

num 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  $\zeta \zeta$ ), o que é um absurdo pois  $i_1, \dots, i_j$  é solução do PCP com listas  $Z$  e  $W$ .

Então:

$$\underbrace{z_{i_1}} \underbrace{z_{i_2}} \dots \underbrace{z_{i_{n+2}}} = \underbrace{w_1} \underbrace{w_{i_2}} \dots \underbrace{w_{n+2}}$$

$$\zeta h_R(x_1) h_R(x_{i_2-1}) \dots \zeta = h_L(y_1) h_L(y_{i_2-1}) \dots \zeta \zeta$$

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

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

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

$\Rightarrow (1, i_2-1, \dots, i_j-1)$  é solução do PCPM com listas  $X$  e  $Y$

Lema 6.2 . Se o PCPM fosse decidível, então o problema da parada de MT 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 :

	LISTA X	LISTA Y	
	#	#q <sub>0</sub> w #	w ∈ Σ*
GRUPO I	x #	x #	x ∈ Γ - {b}
GRUPO II	qx zqx q# zq#	yp pzy yp# pzy#	$\begin{aligned} &\text{se } \delta(q, x) = (p, y, \rightarrow) \\ &\text{se } \delta(q, x) = (p, y, \leftarrow) \\ &\text{se } \delta(q, b) = (p, y, \rightarrow) \\ &\text{se } \delta(q, b) = (p, y, \leftarrow) \end{aligned} $

onde  $x, y, z \in \Gamma - \{b\}$  . As cadeias deste grupo são simulações de transições de  $M$ , onde as cadeias da lista  $Y$  são obtidas a partir das cadeias da lista  $X$  correspondente, em uma transição.

Grupo III

$xqy$   
 $xq\#$   
 $\#qy$   
 $q\#\#$

$q$   
 $q\#$   
 $\#q$   
 $\#$

;  $q \in F$

A ideia é fazer a diferença entre os componentes das listas X e Y ir diminuindo passo a passo (até que eles se igualem), acrescentando mais símbolos à lista X do que à lista Y.

Seja a entrada  $w = x_1 \dots x_m$  e seja

$$q_0 x_1 \dots x_m \vdash \alpha_1 q_1 \beta_1 \vdash \dots \vdash \alpha_k q_k \beta_k \quad ; \quad q_i \notin F \quad (i=0,1,\dots,k-1)$$

a sequência de configurações de M com entrada w. Deve-se mostrar que enquanto M não alcança um estado final, não é possível conseguir uma solução para o PCPM com listas X e Y (a lista Y estará sempre maior que a lista X).

Inicialmente (para o PCPM deve-se ter  $i_1 = 1$ ) tem-se:

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 deve-se colocar  $y_1 q_1$  na lista Y). Para fazer aparecer  $x_2 \dots x_m \#$  na lista X deve-se usar cadeias do grupo I (que serão repetidas na lista Y). Após esse passo, tem-se:

LISTA X:

$\# q_0 x_1 \dots x_m \#$

LISTA Y:

$\# q_0 x_1 \dots x_m \# y_1 q_1 x_2 \dots x_m \#$

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

Considere agora o caso em que  $q_k \in F$ . Nesse ponto as listas serão:

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

e seja  $\alpha_k q_k \beta_k = \alpha' x q_k y \beta$ . Então

LISTA X:  $\# q_0 x_1 \dots x_m \# \dots \# \alpha_{k-1} q_{k-1} \beta_{k-1} \#$

LISTA Y:  $\# q_0 x_1 \dots x_m \# \dots \# \alpha_{k-1} q_{k-1} \beta_{k-1} \# \alpha' x q_k y \beta \#$

Agora, como  $q_k \in F$ , deve-se usar cadeias do grupo III. Deve-se colocar  $\alpha' x q_k y \beta$  em X e  $\alpha' q_k \beta$  em Y (note que a cadeia  $x q_k y$  de X corresponde à cadeia  $q_k$  de 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 entre as listas vai diminuindo (as cadeias do grupo III são tais que acrescenta-se 3 símbolos à lista X (lista menor) e apenas 1 (ou 2) símbolos à lista Y (lista maior)).

Logo, se M atinge um estado final, pode-se igualar as duas



listas usando cadeias do grupo III. Portanto:

M para com entrada  $w \Leftrightarrow$  PCPM com listas  $X$  e  $Y$  tem solução

Teorema 6.2 . O PCP não é decidível

Prova . Consequência dos lemas 6.1 e 6.2

#### APLICAÇÃO DO PCP

Sejam as listas

$$X = (x_1, \dots, x_k)$$

$$Y = (y_1, \dots, y_k)$$

sobre um alfabeto  $\Sigma$ .

Seja  $A = \{a_1, \dots, a_k\}$  tal que  $A \cap \Sigma = \emptyset$ .

Sejam as GLC

$$G_A = (\{S_A\}, T, P_A, S_A)$$

$$G_B = (\{S_B\}, T, P_B, S_B)$$

tais que :  $T = A \cup \Sigma$

$$P_A = \{ S_A \rightarrow x_i S_A a_i \mid x_i a_i \}$$

$$(1 \leq i \leq k)$$

$$P_B = \{ S_B \rightarrow y_i S_B a_i \mid y_i a_i \}$$

Portanto :

$$L_A = L(G_A) = \{ x_{i_1} \dots x_{i_l} a_{i_l} \dots a_{i_1} \mid l \geq 1 \}$$

$$L_B = L(G_B) = \{ y_{i_1} \dots y_{i_l} a_{i_l} \dots a_{i_1} \mid l \geq 1 \}$$

Teorema 6.3. O problema "uma GLC  $G$  é ambígua?" é indecidível.

Prova:

Seja  $G = (\{S, S_1, S_2\}, \tau, P, S)$  onde

$$P = \{ S \rightarrow S_1 \mid S_2, S_1 \rightarrow x_i S_1 a_i \mid x_i a_i, S_2 \rightarrow y_i S_2 a_i \mid y_i a_i \}$$

$(1 \leq i \leq k)$

Então  $L(G) = L_A \cup L_B$

Portanto:

$G$  é ambígua  $\Leftrightarrow$  o PCP com listas  $X$  e  $Y$  possui solução

## BIBLIOGRAFIA

HOPCROFT, J.E.; ULLMAN, J.D. Introduction to automata theory, languages, and computation. Addison-Wesley, Reading, MA, 1979

HOPCROFT, J.E.; ULLMAN, J.D. Formal languages and their relation to automata. Addison-Wesley, Reading, MA, 1969

AHO, A.V.; ULLMAN, J.D. The theory of parsing, translation and compiling, vol. I: Parsing. Prentice-Hall, Englewood Cliffs, NJ, 1972

HARRISON, M.A. Introduction to formal language theory. Addison-Wesley, Reading, MA, 1978.

MINSKY, M.L. Computation: finite and infinite machines. Prentice-Hall, Englewood Cliffs, NJ, 1967

ARBIB, M.A. Theories of abstract automata. Prentice-Hall, Englewood Cliffs, NJ., 1970.

MANNA, Z. Mathematical theory of computation. McGraw-Hill Kogakusha, Toshio, Tokyo, Japan, 1974

JONES, N.D. Computability theory: an introduction. Academic Press, NY, 1973.

BOOTH, T.L. Sequential machines and automata theory. John Wiley, New York, NY, 1967.

ALTO, A.V.; ULLMAN, J.D. Principles of Compiler Design. Addison-Wesley, Reading, MA, 1977