O Problema de Parada

Def: Dada uma M.T. M, denotamos por $c(M)$ o "programa" de M para a máquina universal $\mathcal{U}$

- Considere a linguagem

$$H = \{c(M)\,z\,w : \text{a máquina } M \text{ para com entrada } w \in \Sigma_{\mathcal{U}}^*\}$$

- H é recursivamente enumerável: é fácil construir uma máquina $T_{\Sigma_{\mathcal{U}}}$ que transforma $\triangleright c(M)\,z\,w$ em $\triangleright c(M)\,z\,c(w)$, que podemos rodar em $\mathcal{U}$, e que vai parar porque M para com w.

→ A máquina M_H $T_{\Sigma_{\mathcal{U}}} \to \mathcal{U}$ que roda $T_{\Sigma_{\mathcal{U}}}$ e depois $\mathcal{U}$ é uma máquina que aceita H

- Considere a linguagem

$$H' = \{c(M) : \text{a M.T. } M \text{ para com entrada } c(M) \in \Sigma_{\mathcal{U}}^*\}$$

⤳ não é tão estranho: recompilar um compilador com o próprio compilador

- Como H é R.E., podemos provar que H' também é R.E.

Prova: Considere a máquina $\mathcal{C}$ que transforma $\triangleright c(M)$ em $\triangleright c(M)\,z\,c(M)$ e seja $M_{H'}$ a máquina $\mathcal{C} \to M_H$, que roda $\mathcal{C}$ e depois M_H. Se $c(M) \in H'$, então $M_{H'}$ aceita $c(M)$. Logo, $M_{H'}$ aceita H'

- Finalmente, considere a linguagem $\overline{H'} \subseteq \Sigma_{\mathcal{U}}^*$. Há três tipos de elementos:

(1) $w = c(M)$ e M não para com entrada $c(M)$

(2) $w = c(M)$ e M não processa palavras em $\Sigma_{\mathcal{U}}^*$

(3) w não é programa de uma M.T.

PERGUNTA: $\overline{H'}$ É R.E.?

TEOREMA: $\overline{H'}$ NÃO É R.E.

PROVA: SUPONHA QUE $\overline{H'}$ É R.E. E SEJA N UMA M.T. QUE ACEITA $\overline{H'}$.

PERGUNTA: $c(N) \in H'$ OU $c(N) \in \overline{H'}$?

CASO 1: $c(N) \in H'$. PELA DEFINIÇÃO DE H', N PARA COM ENTRADA $c(N)$.
ISSO IMPLICA QUE $c(N)$ É ACEITA POR N, E PORTANTO,
$c(N)$ ESTÁ NA LINGUAGEM ACEITA POR N, I.E., $\overline{H'}$. UMA CONTRADIÇÃO.

CASO 2: $c(N) \in \overline{H'}$
COMO $c(N)$ É UMA CODIFICAÇÃO DE M.T., $c(N)$ NÃO É DO TIPO (3)
COMO $c(N)$ É A MÁQUINA QUE ACEITA $\overline{H'} \subseteq \Sigma_u^*$,
$c(N)$ NÃO É DO TIPO (2)
LOGO, $c(N)$ É DO TIPO (1), OU SEJA,
$c(N)$ NÃO PARA COM ENTRADA $c(N)$
ISSO IMPLICA QUE $c(N)$ NÃO ESTÁ NA LINGUAGEM ACEITA POR N,
I.E. $c(N) \notin \overline{H'}$. UMA CONTRADIÇÃO.

PORTANTO, NÃO EXISTE M.T. QUE ACEITE $\overline{H'}$. ∎

- COMO $\overline{H'}$ NÃO É R.E., TEMOS QUE $\overline{H'}$ NÃO É REC.

COROLÁRIO: HÁ LINGUAGENS R.E. QUE NÃO SÃO REC.

COROLÁRIO: HÁ LINGUAGENS R.E. CUJO COMPLEMENTO NÃO É R.E.
OU SEJA, O CONJ. DE LINGUAGENS R.E. NÃO É FECHADO
PELA OPERAÇÃO DE COMPLEMENTO.

Problema de Parada: Fixe alfabeto Σ. Dada M.T. M com alfabeto Σ, e $w \in \Sigma^*$, existe M.T. que tendo $c(M)Zw$ como entrada, decide se M para com entrada w?

- Em caso afirmativo, toda linguagem R.E. seria Rec

Prova: Seja L uma linguagem RE e M uma M.T. que aceita L.
Seja A_M a M.T. que transforma $\triangleright w$ em $\triangleright c(M)Zw$.
Se $w \in L$, então M para com entrada w, e $A_M \to P_\Sigma$ para no estado SIM
se $w \in L$, então $A_M \to P_\Sigma$ para no estado NÃO.

$\Rightarrow$ Quando M aceita L, $A_M \to P_\Sigma$ decide L.

- Logo o Problema de Parada não pode ter resposta afirmativa, pois existem linguagens R.E. que não são Rec.

- Nem toda questão matemática pode ser resolvida algoritmicamente

- Outros problemas indecidíveis
  1) Decidir se a interseção de duas LLC é livre de contexto
  2) Decidir se uma LLC é regular
  3) Decidir se uma LLC é inerentemente ambígua

Reduções:

- Método para provar que uma linguagem é indecidível

- Dizemos que L_1 é redutível a L_2 se existe função computável $\tau: \Sigma^* \to \Sigma^*$ t.q. $w \in L_1$ se e somente se $\tau(w) \in L_2$.

- Nesse caso, se L_1 não é Rec, então L_2 não é Rec, do contrário encontraríamos M.T. que decide L_1