Notação: a descrição de uma MT M será denotada por $c(M)$ ou "M"

a descrição de uma entrada w para M será denotada por $c(w)$ ou "w"

Σ_U o alfabeto da máquina universal

Problema da Parada

Existe uma M.T. M (usando o alfabeto universal Σ_U) com a propriedade de que, ao receber como entrada "N""w", sendo N uma M.T. e "w" uma entrada para N, M para aceitando, se N para ao rodar com entrada w, e M para rejeitando, se N não para ao rodar com entrada w?

Em outras palavras, a linguagem $H = \{ "N""w" : N$ é uma M.T. que para ao rodar com entrada $w \}$ é recursiva (decidível)?

Teorema: $H \in R.E. \setminus Rec$

Prova: 1° $H \in R.E.$: quero uma MT. M tal que

Quando executada na entrada "N""w", M:

- para, se N para quando executada com w
- não para, caso contrário

A máquina universal U serve perfeitamente.

2º $H \notin Rec$: Suponha, para contradição, que $H \in Rec$.

Então note que $H' = \{ "N" \in \Sigma_U^* ; "N""N" \in H \}$

$= \{ "N" \in \Sigma_U^* ; N \text{ para com entrada } "N" \}$

$= \{ w \in \Sigma_U^* ; w \text{ descreve M.T. } N \ \& \ N \text{ para com entrada } N \}$

também é recursiva.

Portanto $\overline{H'}$ também é recursiva

$\overline{H'} = \{ w \in \Sigma_U^* ; w \text{ não descreve M.T. ou } (w = "N" \ \& \ N \text{ não para com entrada } "N") \}$

Entretanto:

<u>Afirmação</u>: $\overline{H'}$ não é recursivamente enumerável.

Suponha que $\overline{H'}$ fosse RE, aceita pela M.T. N

(ou seja, $w \in \overline{H'} \iff N$ para com entrada w)

Então $\text{"}N\text{"} \in \overline{H'} \iff N$ para com entrada "N"

$$\overset{def}{\iff} \text{"}N\text{"} \in H',$$

impossível.

Logo $H \notin Rec.$ ∎

2ª ideia: Tabela cujas linhas & colunas são os elementos de Σ^*.

	w_0	w_1	w_2	w_3	w_4
w_0	S	S	N	N	N
w_1	N	S	N	N	N
w_2	N	N	S	N	N
w_3	N	N	N	N	N
w_4	N	N	N	N	S

H

S: w_{linha} descreve M.T. N, w_{coluna} descreve entrada v para N, e N para com a entrada v

N: caso contrário

Fato: As linhas da tabela denotam todos os conjuntos R.E.

Podemos definir $D = \{ w_i ; \text{ na tabela, a célula } (w_i, w_i) \text{ contém } N \}$ $= \overline{H'}$

Logo D não corresponde a nenhuma linha da tabela!

Mas, se a tabela (i.e., H) fosse recursiva, então D seria recursivo, logo R.E., logo uma linha da tabela. ∎

Corolário: R.E. não é fechada para complemento

Prova: Vimos que $\overline{H'} \notin RE$, mas $H' \in RE$ ∎

De posse de uma linguagem específica $H \in R.E. \setminus Rec$, abrem-se possibilidades para provarmos que certas $L \notin Rec$.

Def: Dadas linguagens $L_1, L_2 \in \Sigma_0^*$, dizemos que L_1 é redutível por mapeamento a L_2, denotado $L_1 \leq_m L_2$, se existe $f: \Sigma_0^* \to \Sigma_0^*$ computável tal que

$$w \in L_1 \iff f(w) \in L_2$$ $\Bigg\{$ portanto também $w \notin L_1 \iff f(w) \notin L_2$

(alternativamente, $L_1 = f^{-1}[L_2]$)

Def: $f : \Sigma_0^* \to \Sigma_0^*$ é computável se existe MT $M = (\Sigma_0, \ldots)$ tal que $\forall w \in \Sigma_0^*$ temos que, se M é executada com entrada w, M para em algum momento com $f(w)$ na fita.

Teorema: Se $L_1 \leq_m L_2$ & $L_2 \in Rec$

então $L_1 \in Rec$

Prova: Se M_f computa f (f reduz L_1 a L_2) e

M_2 decide L_2, então a máquina M_1 que:

1) Ao receber w de entrada, usa M_f para computar $f(w)$
2) Depois executa M_2 em $f(w)$

decide L_1. $\blacksquare$

Corolário: se $L_1 \leq_m L_2$ & $L_1 \notin Rec$, então $L_2 \notin Rec$.

Logo: para provar que $L \notin Rec$, basta provar $H \leq_m L$

Exercício: Mostre que são indecidíveis os problemas

a) Dada M.T. M, decidir se M para com a entrada ε

b) " " " " " " " " alguma entrada

$\exists$ entrada w tal que M para com entrada w.

c) " " " " " " " " todas as entradas

d) Dadas MTs M & N, decidir se M & N param exatamente nas mesmas entradas

<u>Soluções</u>: a) Queremos reduzir H a esse problema.

Quero definir f computável tal que

"N""w" $\in H$ $\iff$ f("N""w") é <u>sim</u> para o problema do item (a)

$\iff$ f("N""w") = "N_0" tal que N_0 para quando executada com entrada ε

$f(\text{"}N\text{""}w\text{"})$ é (a descrição de) uma máquina que, ao rodar na entrada vazia, primeiro escreve w na fita, depois simula N.

b) Redução de Parada a (b):

ao receber "N""w", construímos uma máquina de Turing como a seguir:

- Primeiro apaga qualquer entrada recebida
- Escreve w
- Roda N

Logo a nova máquina:

- para com qualquer entrada, se N para com entrada w
- não para com nenhuma entrada, se N não para com entrada w

Portanto a máquina para com alguma entrada
$$\Longleftrightarrow$$
N para com entrada w

c) Mesma coisa da b!

d) Reduzimos (c) a (d):

Dada "M" (para a qual queremos saber se M aceita todas as palavras), deixamos "M" na fita e, ao lado, escrevemos a descrição de uma máquina que aceite qualquer palavra (por exemplo, a máquina que para imediatamente após ser iniciada).

Logo M aceita todas as palavras

$\Longleftrightarrow$

M & (a máquina que aceita todas as palavras) aceitam as mesmas palavras