Resumo

Por $\mathcal{Alef\ Keuffer}$

Exemplos

De problemas de decisão decidíveis

Seja L uma linguagem recursiva. Dada uma palavra w, tem-se $w \in L$

De problemas de decisão indecidíveis

- Dada uma palavra $w \in x.y^*$, tem-se $w \in \mathrm{AutoAceite}$?
  - devido a $\mathrm{AutoAceite}$ ser não recursiva. Não existe algoritmo que decida $\mathrm{AutoAceite}$. No entanto esta linguagem é recursivamente enumeravel. Diz-se então que o problema é semi-decidível (isto significa que existe uma máquina de Turing que permite responder nos casos afirmativos, ou seja, nos casos em que w é uma palavra de $\mathrm{AutoAceite}$).
- $\mathrm{Aceita}_w(\mathcal{T})$ [Aceitação]
  - $\mathrm{AceitaTudo}(\mathcal{T})$
  - $\mathrm{Aceita}_\varepsilon(\mathcal{T})$
- $\mathrm{Para}_w(\mathcal{T})$ [Paragem]
  - $\mathrm{AceitaNada}(\mathcal{T})$
  - $\mathrm{Para}_\varepsilon(\mathcal{T})$
- $\mathrm{Equiv}(\mathcal{T}_1, \mathcal{T}_2)$: "$L(\mathcal{T}_1) = L(\mathcal{T}_2)$"
- $\mathrm{Sub}(\mathcal{T}_1, \mathcal{T}_2)$: "$L(\mathcal{T}_1) \subseteq L(\mathcal{T}_2)$"
- "$L(\mathcal{T}_1) \cap L(\mathcal{T}_2) = \emptyset$"

De linguagens não recursivamente enumeráveis

- $\mathrm{NãoAutoAceite}$

De linguagens recursivamente enumeráveis não recursivas

- $\mathrm{AutoAceite}$

De problemas indecidíveis sobre linguagens recursivamente enumeráveis

- $\varepsilon \in L$
- $L = \emptyset$
- $L = A^*$

De prova usando teorema de Rice

Diga se a afirmação é verdadeira ou falsa justificando.

O problema "Dada uma máquina de Turing $\mathcal{T}$, será que $L(\mathcal{T}) \subseteq a^*$?" e decidıvel

Seja $d = L(\mathcal{T})$

Note que $d \in D = \{L : L \text{ é uma linguagem recursivamente enumerável}\}$

Seja $P(x)$: "$x \subseteq a^*$" para $x \in D$.

Note que P não é trivial porque $a^*, b^* \in D$ mas $P(a^*)$ é verdade e $P(b^*)$ é falso.

Logo, pelo Teorema de Rice, P é indecidível.

Observações

- $\mathrm{AutoAceite}$ contém palavras que codificam máquinas de Turing que reconhecem sua codificação.

Máquinas Auxiliares

- $\mathrm{Escreve}_w$
- $\mathrm{ApagaFita}$

Definições

Função característica

Seja L uma linguagem sobre um alfabeto A. A função característica de L é a função

$$\chi_L : A^* \to \{0,1\}$$

definida para cada $u \in A^*$, por

$$\chi_L(u) = \begin{cases} 1 \text{ se } u \in L \\ 0 \text{ se } u \notin L \end{cases}$$

Definição 1

Seja $L \subseteq A^*$ uma linguagem e seja $\mathcal{T}$ uma máquina de Turing com alfabeto de entrada A. Diz-se que

- $\mathcal{T}$ **aceita** ou **reconhece** L se $L = L(\mathcal{T})$.
- $\mathcal{T}$ **decide** L se a função característica χ_L é calculada por $\mathcal{T}$.

Definição 2

Uma linguagem L diz-se

- **recursivamente enumerável** se existe uma MT que reconhece L.
- **recursiva** (ou **decidivel**) se existe uma MT que decide L.

Função Codificadora

$$\begin{aligned} c : \mathrm{MT_N} &\to \{x,y\}^* \\ \mathcal{T} &\mapsto c(\mathcal{T}) \end{aligned}$$

- $c'(q_i) = c'(s_i) = x^{i+1}$
- $c'(C) = x$, $c'(E) = x * 2$, $c'(D) = x^3$

Note-se em particular,

- $c'(\Delta) = c'(s_0) = x$ e $c'(f) = c'(q_0) = x$

A cada transição e, descrita por $\delta(q,t) = (q', t', m)$

$$c'(e) = c'(q)yc'(t)yc'(q')yc'(t')yc'(m)y$$

Depois, codifica-se a máquina de Turing $\mathcal{T}$ pela palavra

$$c(\mathcal{T}) = c'(q_i)yc'(e_1)yc'(e_2)\cdots yc'(e_k)y$$

onde q_i é o estado inicial de $\mathcal{T}$ e $e_1, e_2, \ldots, e_k$ são as transições de $\mathcal{T}$ numa ordem fixada previamente.

Pode também codificar-se cada palavra $w = r_1r_2\cdots r_n$, onde $r_i \in \mathcal{S}$, por

$$c(w) = yyc'(r_1)yc'(r_2)\cdots yc'(r_n)y$$

Quando se considera uma sequência

$$c(\mathcal{T})c(w) = c'(q_i)yc'(e_1)yc'(e_2)\cdots yc'(e_k)yyyc'(r_1)yc'(r_2)\cdots yc'(r_n)y$$

fica claro onde $c(\mathcal{T})$ termina devido ao prefixo yy de $c(w)$.

Exemplo

$$c(\mathcal{T}) = \underbrace{x^2}_{c'(q_1)}y\underbrace{x^2yxyx^3y}_{c'(e_1)}y\underbrace{x^3yx^2yx^3yx^3yx^3y}_{c'(e_2)}y\cdots$$

Proposicões e Teoremas

Proposição 1. Sejam L e K linguagens sobre um alfabeto A

- Se L e K são recursivas (resp. recursivamente enumeráveis), então $L \cup K$ e $L \cap K$ são recursivas (resp. recursivamente enumeráveis).
- Se L é recursiva, então $\overline{L}$ e recursiva.

Teorema [Post, 1943]. Uma linguagem L é recursiva se e só se L e $\overline{L}$ são recursivamente enumeráveis.

Proposição 2. Sejam P e P' dois problemas de decisão tais que $P \leq P'$

- Se P' é decidível, então P é decidível.
- Se P é indecidível, então P' é indecidível.
- Se P' e semi-decidível, então P e semi-decidível.

Teorema [Rice, 1953]. Se P é uma propriedade não trivial sobre linguagens recursivamente enumeráveis, então P é indecidível.

Convenções

Convenção 1. Assume-se que existem dois conjuntos enumeráveis

$$\mathcal{Q} = \{q_0, q_1, \ldots\} \quad \text{e} \quad \mathcal{S} = \{s_0, s_1, \ldots\}$$

tais que, para cada máquina de Turing,

$$\mathcal{T} = (Q, A, T, \delta, i, f, \Delta)$$

se tem

- $Q \subseteq \mathcal{Q}$, com $f = q_0$
- $T \subseteq \mathcal{S}$, com $\Delta = s_0$

Diz-se que $\mathcal{T}$ é normalizada se todos os estados e todos os símbolos não brancos de $\mathcal{T}$ pertencem a alguma transição.