Máquina de Turing Universal

S. C. Coutinho

Objetivo

Descrever uma máquina de Turing $\mathcal{U}$, capaz de simular qualquer outra máquina de Turing M. Para isto a máquina deve conter na fita:

- o conjunto de instruções sobre o comportamento da máquina a ser simulada;
- o conteúdo da fita da máquina a ser simulada.

O alfabeto de $\mathcal{U}$

$$\Sigma = \{\triangleright, 0, 1, \sigma, q, X, Y, Z, \#, \sqcup, a, b\}.$$

Este alfabeto é usado para descrever tanto os estados, quanto os símbolos da máquina M. Para isso:

- enumeramos, separadamente, os símbolos e os estados de M;
- para distinguir símbolos de estados, adicionamos aos símbolos o prefixo σ e aos estados o prefixo q;
- representamos as setas $\longrightarrow$ e $\longleftarrow$ como se fossem símbolos de M.

A máquina M

Digamos que, usando a máquina $\mathcal{U}$, desejamos simular uma máquina M que tem

- n estados e
- um alfabeto com m símbolos.

Os estados e os símbolos de M serão enumerados em unário na fita de $\mathcal{U}$. Isto é, o número n será representado por

$$000 \cdots 0 = 0^n$$

uma seqüência de n zeros.

Codificando símbolos

- Há um total de m símbolos a serem representados;
- mais dois "extras", correspondentes a $\rightarrow$ e $\leftarrow$.
- A primeira casa na representação de cada símbolo será sempre ocupada por σ.
- $\triangleright$, $\rightarrow$ e $\leftarrow$ serão sempre os três primeiros símbolos a serem enumerados.
- A representação de um símbolo ocupa no máximo $m+3$ casas. Se menos de $m+3$ casas estiverem sendo ocupadas, as restantes serão preenchidas por 1's.

Exemplos

Por exemplo, o k-ésimo símbolo aparece como

$$\sigma\,\underbrace{000\cdots0}_{k}\underbrace{1\cdots1}_{n-k} = \sigma 0^k 1^{n-k}$$

Lembrando que

$$\triangleright = \sigma 0 1^{m+1}$$

$$\rightarrow = \sigma 0^2 1^m$$

$$\leftarrow = \sigma 0^3 1^{m-1}$$

Codificando estados

- Para cada estado serão alocadas $n + 1$ casas.
- A primeira casa será preenchida por q.
- As casas seguintes serão preenchidas por k zeros, se se trata do k-ésimo estado.
- As casas restantes serão completadas por 1's.

Por exemplo, o k-ésimo estado aparece como

$$q\,\underbrace{000\cdots 0}_{k}\,\underbrace{1\cdots 1}_{n-k} = q0^k1^{n-k}$$

Codificando transições

A transição

$$\delta(q_i, \sigma_j) = (q_r, \sigma_s)$$

de M é codificada na fita de $\mathcal{U}$ na forma

X	$q0^i1^{n-i}$	$\sigma 0^j1^{m+2-j}$	$q0^r1^{n-r}$	$\sigma 0^s1^{m+2-s}$	X

Note que o $q0^i1^{n-i}$ representa na verdade um segmento de $n+1$ casas da fita, e assim por diante.

A fita padrão de $\mathcal{U}$

Na descrição completa de M com entrada w na fita de $\mathcal{U}$ abaixo, S_i denota um segmento de transição, e sempre aparece entre dois Xs.

$\triangleright$	X	S_1	X	$\cdots$	S_t	Y	Q	Z	σ	u_1	$\#$	u_2	$\cdots$

Nesta fita temos que:

De	**Até**	**Conteúdo do segmento**
$\triangleright$	Y	comportamento de M
Y	Z	Q é o estado atual de M
Z	final	w

A $\#$ indica a posição atual do cabeçote na fita de M.

Exemplo

A máquina M tem
Alfabeto: $\{0, \sqcup, \triangleright\}$

Estados: $\{q_0, q_1, h\}$

Os símbolos e estados de M são codificados como abaixo:

estado	código	símbolo	código
q_0	$q01^2$	$\triangleright$	$\sigma01^4$
q_1	$q0^21$	$\rightarrow$	$\sigma0^21^3$
h	$q0^3$	$\leftarrow$	$\sigma0^31^2$
*	*	$\sqcup$	$\sigma0^41$
*	*	0	$\sigma0^5$

Exemplo

Tabela de transição:

estado	entrada	transições
q_0	0	$(q_1, \sqcup)$
	$\sqcup$	$(h, \sqcup)$
	$\triangleright$	$(q_0, \longrightarrow)$
q_1	0	$(q_0, 0)$
	$\sqcup$	$(q_0, \longrightarrow)$
	$\triangleright$	$(q_1, \longrightarrow)$

A primeira transição é codificada como:

X	$q01^2$	$\sigma 0^5$	$q0^21$	$\sigma 0^41$	X

A linguagem

- M = máquina de Turing M no alfabeto Σ;
- $c(M) \in \Sigma^*$ a representação de M na fita de $\mathcal{U}$ (trecho entre o primeiro X e o Z).

Defina:

$$\mathcal{L} = \{c(M) : M \text{ é uma máquina de Turing no alfabeto } \Sigma\}.$$

e também

$$\mathcal{L}_0 = \{c(M) : \text{a máquina } M \textit{ aceita } c(M)\} \subset \mathcal{L}.$$

$\mathcal{L}_0$ é recursivamente enumerável

Seja $\mathcal{A}$ a máquina de Turing que, tendo como entrada

$$\triangleleft w \sqcup$$

produz a saída

$$\triangleleft w Z w \sqcup .$$

Então:

a máquina $\mathcal{A} \cdot \mathcal{U}$ aceita $\mathcal{L}_0$.

Se $w \in \Sigma^*$ não é uma descrição legítima de alguma máquina de Turing, então wZw é automaticamente rejeitada por $\mathcal{U}$ no momento da checagem inicial da entrada.

Pergunta

$\mathcal{L}_0$ é uma linguagem recursiva?

Lembrete: Se $\mathcal{L}_0$ for recursiva, seu complemento $\overline{\mathcal{L}_0}$ também será.

Teorema

$\overline{\mathcal{L}_0}$ não é sequer recursivamente enumerável!

Demonstração

Por contradição, suponha que $\overline{\mathcal{L}_0}$ é aceita por uma máquina de Turing M no alfabeto Σ.

Logo, o código de M é um elemento de $\mathcal{L}$.

Pergunta:

$$c(M) \text{ pertence a } \overline{\mathcal{L}_0}?$$

Se a resposta fosse SIM

$$c(M) \in \overline{\mathcal{L}_0} = L(M).$$

Se a resposta fosse SIM

$$c(M) \in \overline{\mathcal{L}_0} = L(M).$$

$$\Updownarrow$$

$$c(M) \in L(M)$$

Se a resposta fosse SIM

$$c(M) \in \overline{\mathcal{L}_0} = L(M).$$

$$\Updownarrow$$

$$c(M) \in L(M)$$

$$\Updownarrow$$

$c(M)$ é aceita por M

Se a resposta fosse SIM

$$c(M) \in \overline{\mathcal{L}_0} = L(M).$$

$$\Updownarrow$$

$$c(M) \in L(M)$$

$$\Updownarrow$$

$$c(M) \text{ é aceita por } M$$

$$\Updownarrow$$

$$c(M) \in \mathcal{L}_0 \text{ pela definição de } \mathcal{L}_0$$

Se a resposta fosse SIM

$$c(M) \in \overline{\mathcal{L}_0} = L(M).$$

$$\Updownarrow$$

$$c(M) \in L(M)$$

$$\Updownarrow$$

$c(M)$ é aceita por M

$$\Updownarrow$$

$c(M) \in \mathcal{L}_0$ pela definição de $\mathcal{L}_0$

$$\Downarrow$$

CONTRADIÇÃO

Se a resposta fosse NÃO

$$c(M) \notin \overline{\mathcal{L}_0} = L(M).$$

Se a resposta fosse NÃO

$$c(M) \notin \overline{\mathcal{L}_0} = L(M).$$

$$\Updownarrow$$

$$c(M) \notin L(M)$$

Se a resposta fosse NÃO

$$c(M) \notin \overline{\mathcal{L}_0} = L(M).$$

$$\Updownarrow$$

$$c(M) \notin L(M)$$

$$\Updownarrow$$

$c(M)$ não é aceita por M

Se a resposta fosse NÃO

$$c(M) \notin \overline{\mathcal{L}_0} = L(M).$$

$$\Updownarrow$$

$$c(M) \notin L(M)$$

$$\Updownarrow$$

$c(M)$ não é aceita por M

$$\Updownarrow$$

$c(M) \notin \mathcal{L}_0$ pela definição de $\mathcal{L}_0$

Se a resposta fosse NÃO

$$c(M) \notin \overline{\mathcal{L}_0} = L(M).$$

$$\Updownarrow$$

$$c(M) \notin L(M)$$

$$\Updownarrow$$

$c(M)$ não é aceita por M

$$\Updownarrow$$

$c(M) \notin \mathcal{L}_0$ pela definição de $\mathcal{L}_0$

$$\Downarrow$$

CONTRADIÇÃO

Conseqüência

O complemento de uma linguagem recursivamente enumerável não é necessariamente recursivamente enumerável.

O problema da parada

Dada uma máquina de Turing M no alfabeto Σ e uma palavra $w \in \Sigma^*$ existe uma máquina de Turing $\mathcal{P}$ que, tendo $c(M) \cdot Z \cdot w$ como entrada decide se M pára com entrada w?

NÃO!

- M é uma máquina que aceita $\mathcal{L}_0$;
- $\mathcal{C}$ a máquina que converte

$$\triangleleft w \sqcup \quad \text{em} \quad \triangleleft\, c(M) Z w \sqcup .$$

Portanto,

- Se $w \in \mathcal{L}_0$ então a máquina M pára na entrada w, de modo que $\mathcal{CP}$ pára no estado *sim*.

- Se $w \notin \mathcal{L}_0$, então M não pára com entrada w; mas neste caso $\mathcal{CP}$ também pára, só que no estado *não*.

Só que $\mathcal{L}_0$ não é recursiva!

Outros problemas sem solução

- Dadas duas linguagens livres de contexto, determinar se sua intersecção é livre de contexto.
- Dada uma linguagem livre de contexto, determinar se é regular.
- Dada uma linguagem livre de contexto, determinar se é inerentemente ambígüa.