Teoria da Computação
Indecidibilidade

Leonardo Takuno
{leonardo.takuno@gmail.com}

Centro Universitário Senac

Sumário

1. Máquina de Turing Universal
2. O problema da Parada
3. Linguagem Turing-irreconhecível
4. Método da Diagonalização

Sumário

1. Máquina de Turing Universal
2. O problema da Parada
3. Linguagem Turing-irreconhecível
4. Método da Diagonalização

Máquina de Turing Universal

Sabemos que as linguagens:

$$A_{AFD} = \{\langle B, w\rangle | B \text{ é um AFD que aceita } w\}$$

$$A_{GLC} = \{\langle G, w\rangle | G \text{ é uma GLC que gera } w\}$$

são Turing-decidíveis.
E sobre a linguagem

$$A_{MT} = \{\langle M, w\rangle | M \text{ é uma MT que aceita } w\}.$$

Existe uma máquina de Turing capaz de simular outras Máquinas de Turing ?

Probema de aceitação para Máquinas de Turing

A linguagem

$$A_{MT} = \{\langle M, w\rangle | M \text{ é uma MT que aceita } w\}$$

é chamada de **problema de aceitação para MT** ou **problema da parada** (*the halting problem*).

Máquina de Turing Universal

Teorema: A linguagem A_{MT} é Turing-reconhecível.

Dada a entrada $\langle M, w \rangle$, onde M é uma MT e w é uma cadeia, podemos simular M sobre w?
Podemos simular via máquina de Turing Universal U.
U = "Sobre a entrada $\langle M, w \rangle$, onde M é uma MT e w é uma cadeia:

1. Simule M sobre a entrada w.
2. Se M em algum momento entra no seu estado de aceitação, *aceite*; se M em algum momento entra em seu estado de rejeição, *rejeite*. "

Problema da Parada

A existência da máquina de Turing Universal U mostra que

$$A_{MT} = \{\langle M, w\rangle | M \text{ é uma MT que aceita } w\}.$$

é Turing-reconhecível, mas também podemos decidí-lo?

- O problema ocorre nos casos em que M não pára sobre um determinado w.
- Veremos que este é um insuperável problema: em geral, não se pode decidir se uma MT irá parar sobre w ou não, assim A_{MT} é indecidível.

Sumário

1 Máquina de Turing Universal
2 O problema da Parada
3 Linguagem Turing-irreconhecível
4 Método da Diagonalização

Problema da Parada

Teorema 4.11: A linguagem

$$A_{MT} = \{\langle M, w\rangle | M \text{ é uma MT que aceita } w\}$$

é indecidível.

Prova: Por contradição. Assuma que exista um decisor H para a linguagem A_{TM}. Então,

$$H(\langle M, w\rangle) = \begin{cases} \text{aceite} & \text{se } M \text{ aceita w} \\ \text{rejeite} & \text{se não } M \text{ aceita w} \end{cases}$$

Problema da Parada

Construímos uma nova MT D tal que
D = “Sobre a entrada $\langle Q \rangle$ onde Q é uma MT:

1. Execute H sobre a entrada $\langle Q, \langle Q \rangle \rangle$.
2. Faça o oposto da saída de H; se H aceita, rejeite e se H rejeita, aceite.”

Agora,

$$D(\langle M \rangle) = \begin{cases} \text{aceite} & \text{se } M \text{ não aceita } \langle M \rangle \\ \text{rejeite} & \text{se } M \text{ aceita } \langle M \rangle \end{cases}$$

Mas ...

Problema da Parada

O que acontece quando rodamos D com sua própria descrição $\langle D \rangle$ como entrada ? Nesse caso, obtemos

$$D(\langle D \rangle) = \begin{cases} \text{aceite} & \text{se } D \text{ não aceita } \langle D \rangle \\ \text{rejeite} & \text{se } D \text{ aceita } \langle D \rangle \end{cases}$$

- Independentemente do que D faz, ela é forçada a fazer o oposto.
- Isto é uma contradição. Portanto, nem H nem D pode existir e A_{MT} é indecidível.

Linguagem Turing-irreconhecível

A indecidibilidade de A_{MT} tem profundas ramificações e implica na existência de linguagens que não são computáveis, isto é, linguagens que não são Turing-reconhecíveis. Tal linguagem é o complemento de A_{MT}.

Teorema: Uma linguagem é decidível sse ela é Turing-reconhecível e co-Turing-reconhecível

- Sabemos que A_{MT} é Turing-reconhecível pois utilizamos uma máquina universal para reconhecê-la.
- Sabemos que A_{MT} é indecidível, pelo problema da Parada.
- Logo, A_{MT} não é co-Turing-reconhecível.

Sumário

1. Máquina de Turing Universal
2. O problema da Parada
3. Linguagem Turing-irreconhecível
4. Método da Diagonalização

Linguagem Turing-irreconhecível

Teorema: A linguagem $\overline{A_{MT}}$ não é Turing-reconhecível.

Prova: Por contradição. Observe que A_{MT} é indecidível. Além disso, observe que A_{MT} é Turing-reconhecível. Agora, assuma que that $\overline{A_{MT}}$ é também Turing-reconhecível. Verifique que a string w, ou é um elemento de A_{MT}, ou um elemento de $\overline{A_{MT}}$ podemos contruir o seguinte decisor para A_{MT}.

Linguagem Turing-irreconhecível

Seja $M1$ e $M2$ reconhecedores para A_{MT} e $\overline{A_{MT}}$, respectivamente:
M = "Sobre a entrada w, onde w é uma string:

1. Execute $M1$ e $M2$ em paralelo sobre w.
2. Se $M1$ aceita, aceite; se $M2$ aceita, rejeite."

Note que esta máquina é um decisor pois irá parar sobre toda entrada w. Note também, que este decisor contradiz nosso teorema que A_{MT} é indecidível. Portanto, nossa hipótese que $\overline{A_{MT}}$ é Turing-reconhecível deve estar errada. Isto mostra que $\overline{A_{MT}}$ não é Turing-reconhecível.$\square$

Sumário

1. Máquina de Turing Universal
2. O problema da Parada
3. Linguagem Turing-irreconhecível
4. Método da Diagonalização

Método da Diagonalização

- Acabamos de ver uma prova não construtiva de que existem linguagens que não são Turing-reconhecíveis.
- Vamos ver uma prova construtiva que mostra que algumas linguagens não são computáveis por algoritmos.
- Esta prova mostra que o conjunto de todas as máquinas de Turing é contável ao passo que o conjunto de todas as linguagens é incontável.
- Portanto, existem algumas linguagem que não são reconhecíveis por máquina de Turing.

Método da Diagonalização

- Georg Cantor (1873).
- Medida do tamanho de conjuntos infinitos.
- Se tivermos dois conjuntos infinitos, como podemos dizer se um é maior que o outro ou se eles têm o mesmo tamanho?

Método da Diagonalização

- Considere o conjunto $f : A \rightarrow B$
  - f é **um-para-um** se $f(a) \neq f(b)$ sempre que $a \neq b$.
  - f é **sobrejetora** se para todo $b \in B$ existe um $a \in A$ tal que $f(a) = b$.
  - f é **correspondência** se f é tanto um-para-um quanto sobrejetora.
  - Digamos que A e B são de **mesmo tamanho** se existe uma correspondência $f : A \rightarrow B$.

Método da Diagonalização

Exemplo: $f : \mathcal{N} \rightarrow \mathcal{E}$, onde $\mathcal{N}$ é o conjunto dos números naturais e $\mathcal{E}$ é o conjunto dos naturais pares.

- $f(n) = 2n$ é uma correspondência.

n	$f(n)$
1	2
2	4
3	6
$\vdots$	$\vdots$

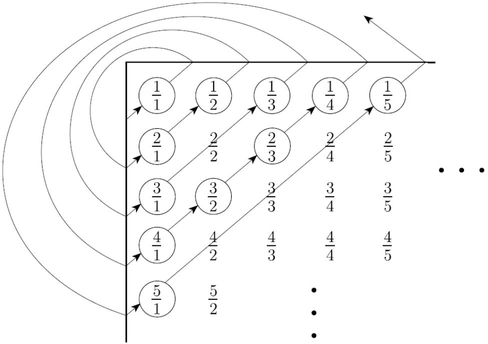
Portanto, declaramos que estes dois conjuntos são do mesmo tamanho.

Definição: Um conjunto é **contável** se é finito ou tem o mesmo tamanho que $\mathcal{N}$

Método da Diagonalização

Teorema: Seja $Q = \{\frac{m}{n} | m, n \in \mathcal{N}\}$ o conjunto dos números racionais positivos, Q é contável.

Idéia da prova:



Método da Diagonalização

Teorema: Seja $\mathcal{R}$ o conjunto dos números reais, $\mathcal{R}$ é incontável.

Prova: Prova por contradição. Assuma que exista uma correspondência f entre $\mathcal{N}$ e $\mathcal{R}$. Agora obtemos um $x \in \mathcal{R}$ que não é pareada com qualquer elemento de $\mathcal{N}$. Escolha o *i-ésimo* dígito fracionário de x diferente do dígito da *i-ésima* fração.

n	$f(n)$
1	$3,\underline{1}4159\ldots$
2	$55,5\underline{5}555\ldots$
3	$0,12\underline{3}45\ldots$
4	$0,500\underline{0}00\ldots$
$\vdots$	$\vdots$

Ex: $x = 0,4641\ldots$. Então, $x \neq f(n)$ para todo n. Logo $\mathcal{R}$ é incontável.$\square$

Uma prova construtiva

Teorema: O conjunto de todas as máquinas de Turing é contável.

Prova: Procedemos por construir uma correspondência entre o número natural e o conjunto de todas as máquinas de Turing.
Observe que o conjunto de todas as strings sobre algum alfabeto é contável, isto é, toda string de um tamanho particular tem um número finito de permutações de símbolos de alfabeto. Isto é, podemos construir uma correspondência de números naturais para strings.
Além disso, note que podemos codificar todos os alfabetos possíveis usando o alfabeto Σ. Isto implica que qualquer máquina de Turing M pode ser codificada com uma string $\langle M \rangle$. Seja $\langle M \rangle^*$ o conjunto de todas as descrições de máquinas de Turing válidas.
Observe que $\langle M \rangle^* \subseteq \Sigma^*$. Sabemos que Σ^* é contável, logo, $\langle M \rangle^*$ é contável. $\square$

Uma prova construtiva

Teorema: O conjunto de todas as seqüências binárias infinita é incontável.

Prova: Prova por contradição. Assuma que podemos construir uma correspondência $f : N \rightarrow \mathcal{B}$, onde $\mathcal{N}$ são os números naturais e $\mathcal{B}$ é o conjunto de todas as seqüências binárias infinitas. Então,

n	$f(n)$
1	0100111 ...
2	11111000 ...
3	1011001 ...
$\vdots$	$\vdots$
n	0010 ... 0_n ...
$\vdots$	$\vdots$

Uma prova construtiva

Então,

n	$f(n)$
1	0100111 ...
2	11111000 ...
3	1011001 ...
$\vdots$	$\vdots$
n	0010 ... 0_n ...
$\vdots$	$\vdots$

Observe que podemos construir uma seqüência binária que difere de todas as seqüências enumeradas pelo menos em 1 bit, 100...1.... Isto é, para qualquer valor de k temos que construir uma seqüência que difere no valor de $f(k)$ no *k-ésimo* bit. Portanto, existem elementos em $\mathcal{B}$ que não são imagens de f. Isto significa nossa hipótese de que f é uma correspondência incorreta.$\square$

Uma prova construtiva

Teorema: O conjunto de todas as linguagens $\mathcal{L}$ sobre o alfabeto $\Sigma_{0,1}$ é incontável.

Prova: Mostramos uma correspondência $f : \mathcal{L} \rightarrow \mathcal{B}$. Para cada linguagem $A \in \mathcal{L}$ podemos construir um único elemento em $\mathcal{B}$. Seja $\Sigma^* = \{s_1, s_2, s_3, ...\}$. O *i-ésimo* bit da **seqüência característica** de A é 1 se $s_i \in A$ e 0 se $s_i \notin A$.

Uma prova construtiva

Teorema: O conjunto de todas as linguagens $\mathcal{L}$ sobre o alfabeto $\Sigma_{0,1}$ é incontável.

Exemplo: se A fosse a linguagem de todas as cadeias começando com 0 sobre o alfabeto $\{0, 1\}$, sua seqüência característica $\mathcal{X}_\mathcal{A}$ seria

$$
\begin{array}{lllllllllll}
\Sigma^* & = \{ & \varepsilon, & 1, & 00, & 01, & 10, & 11, & 000, & 001, & ...\} \\
\mathcal{A} & = \{ & , & , & 00, & 01, & , & , & 000, & 001, & ...\} \\
\mathcal{X}_\mathcal{A} & = \{ & 0\ , & 0, & 1, & 1, & 0, & 0, & 1, & 1\ , & ...\}
\end{array}
$$

A função $f : \mathcal{L} \to \mathcal{B}$, onde $f(A)$ é a seqüência característica de A, é um-para-um e sobrejetora e, portanto, uma correspondência. Conseqüentemente, como $\mathcal{B}$ é incontável, $\mathcal{L}$ também é incontável.$\square$

Uma prova construtiva

Teorema: Algumas linguagens não são Turing-reconhecíveis.

Prova: Observe que o conjunto de Máquinas de Turing é contável e da prova anterior segue que o conjunto de todas as linguagens é incontável. Assim, não conseguimos construir uma correspondência entre o conjunto de todas as linguagens e o conjunto de máquinas de Turing.
Portanto, existem algumas linguagens que não são reconhecidas por Máquina de Turing. $\square$

Exercícios

1) Mostre que o conjunto dos números naturais pares $\{2, 4, 6, \cdots\}$ e o conjunto dos números naturais ímpares $\{1, 3, 5, \cdots\}$ têm o mesmo tamanho.

Exercícios (Solução)

Basta realizar o seguinte emparelhamento (função bijetora) entre os dois conjuntos:

Pares	Ímpares
0	1
2	3
4	5
6	7
8	9
10	11
...	...

A função que mapeia os pares nos ímpares é dada por $f : pares \rightarrow impares$ $f(x) = x + 1$ que, claramente, é uma função bijetora (gráfico é uma reta)