Automi e Linguaggi Formali

Parte 15 – Indecidibilità

UNIVERSITÀ DEGLI STUDI DI PADOVA

Davide Bresolin
Ultimo aggiornamento: 10 maggio 2021

“Questa frase è falsa.”

Un risultato fondamentale

"Esiste un problema specifico che è algoritmicamente irrisolvibile"

- Problemi di interesse non solo teorico, ma anche pratico
- Esempio: Verifica del software
  - verificare che un programma è corretto non è risolvibile algoritmicamente

Sommario

Teorema: A_{TM} è indecidibile

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ è una TM che accetta la stringa } w\}$$

- Chiarimento: A_{TM} è Turing-riconoscibile
- Conseguenza: i riconoscitori sono più potenti dei decisori

Teorema: A_{TM} è indecidibile

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ è una TM che accetta la stringa } w\}$$

- Chiarimento: A_{TM} è Turing-riconoscibile
- Conseguenza: i riconoscitori sono più potenti dei decisori
- $U =$ "Su input $\langle M, w\rangle$, dove M è una TM e w una stringa:
  1. Simula M su input w
  2. Se la simulazione raggiunge lo stato di accettazione, accetta; se raggiunge lo stato di rifiuto, rifiuta."
- U è un riconoscitore. Perché non è un decisore?

Macchina Universale di Turing

- U è un esempio di Macchina Universale di Turing
- Introdotta da Alan Turing nel 1936
- Può simulare qualsiasi macchina di Turing a partire dalla sua descrizione

Sommario

Il metodo della diagonalizzazione

- Dimostrare che A_{TM} è indecidibile usa la diagonalizzazione
- Metodo scoperto da Cantor nel 1873
- Serve per confrontare le dimensioni di insiemi infiniti

Il metodo della diagonalizzazione

- Dimostrare che A_{TM} è indecidibile usa la diagonalizzazione
- Metodo scoperto da Cantor nel 1873
- Serve per confrontare le dimensioni di insiemi infiniti

Idea: due insiemi finiti hanno la stessa dimensione se gli elementi di un insieme possono essere accoppiati agli elementi dell'altro insieme.

Corrispondenze

- Abbiamo due insiemi A e B e una funzione $f : A \mapsto B$
- f è iniettiva se non mappa mai elementi diversi nello stesso punto: $f(a) \neq f(b)$ ogniqualvolta che $a \neq b$
- f è suriettiva se tocca ogni elemento di B: per ogni $b \in B$ esiste $a \in A$ tale che $f(a) = b$
- Una funzione iniettiva e suriettiva è chiamata biettiva: è un modo per accoppiare elementi di A con elementi di B

Corrispondenze

- Abbiamo due insiemi A e B e una funzione $f : A \mapsto B$
- f è iniettiva se non mappa mai elementi diversi nello stesso punto: $f(a) \neq f(b)$ ogniqualvolta che $a \neq b$
- f è suriettiva se tocca ogni elemento di B: per ogni $b \in B$ esiste $a \in A$ tale che $f(a) = b$
- Una funzione iniettiva e suriettiva è chiamata biettiva: è un modo per accoppiare elementi di A con elementi di B

Definition

A e B hanno la stessa cardinalità se esiste una funzione biettiva $f : A \mapsto B$

Esempio

- $\mathbb{N} = \{0, 1, 2, \ldots, \}$, insieme dei numeri naturali
- $\mathbb{E} = \{0, 2, 4, \ldots, \}$, insieme dei numeri pari

Quale dei due insiemi è il più grande?

Esempio: naturali vs numeri pari

Esempio

- $\mathbb{N} = \{0, 1, 2, \ldots, \}$, insieme dei numeri naturali
- $\mathbb{E} = \{0, 2, 4, \ldots, \}$, insieme dei numeri pari

Quale dei due insiemi è il più grande?

Definition (Insieme numerabile)

Un insieme è numerabile se è finito oppure ha la stessa cardinalità di $\mathbb{N}$

Esempi

- $\mathbb{Q}$ è numerabile?
- $\mathbb{R}$ è numerabile?
- Dato un alfabeto finito Σ, Σ^* è numerabile?
- L'insieme di tutte le macchine di Turing è numerabile?
- L'insieme di tutte le sequenze binarie infinite è numerabile?
- Dato un alfabeto finito Σ, l'insieme di tutti i linguaggi su Σ^* è numerabile?

Corollario

- L'insieme di tutte le macchine di Turing è numerabile
- L'insieme di tutti i linguaggi è non numerabile
- Devono esistere linguaggi non riconoscibili da una macchina di Turing

Conclusioni?

Università degli Studi di Padova

No, la lezione non è ancora finita.
Non abbiamo dimostrato che A_{TM} è indecidibile!

Sommario

UNIVERSITÀ DEGLI STUDI DI PADOVA

Teorema: A_{TM} è indecidibile

Università degli Studi di Padova

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ è una TM che accetta la stringa } w\}$$

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ è una TM che accetta la stringa } w\}$$

Dimostrazione:

- per contraddizione. Assumiamo A_{TM} decidibile per poi trovare una contraddizione
- Supponiamo H decisore per A_{TM}
- Cosa fa H con input $\langle M, w\rangle$?

$$H(\langle M, w\rangle) = \begin{cases} \text{accetta} & \text{se } M \text{ accetta } w \\ \text{rifiuta} & \text{se } M \text{ non accetta } w \end{cases}$$

Teorema: A_{TM} è indecidibile

- Definiamo una TM D che usa H come subroutine
- $D =$ "Su input $\langle M \rangle$, dove M è una TM:
  1. Esegue H su input $\langle M, \langle M \rangle \rangle$
  2. Dà in output l'opposto dell'output di H. Se H accetta, rifiuta; se H rifiuta, accetta."
- Cosa fa H con input $\langle D \rangle$?

Teorema: A_{TM} è indecidibile

- Definiamo una TM D che usa H come subroutine
- $D =$ "Su input $\langle M \rangle$, dove M è una TM:
  1. Esegue H su input $\langle M, \langle M \rangle \rangle$
  2. Dà in output l'opposto dell'output di H. Se H accetta, rifiuta; se H rifiuta, accetta."
- Cosa fa H con input $\langle D \rangle$?

$$D(\langle D \rangle) = \begin{cases} \text{accetta} & \text{se } D \text{ non accetta } \langle D \rangle \\ \text{rifiuta} & \text{se } D \text{ accetta } \langle D \rangle \end{cases}$$

- Contraddizione!

Comprendere la dimostrazione

1. H accetta $\langle M, w \rangle$ esattamente quando M accetta w
   a. Banale: abbiamo assunto che H esista e decida A_{TM}
   b. M rappresenta qualsiasi TM e w è una qualsiasi stringa
2. D rifiuta $\langle M \rangle$ esattamente quando M accetta $\langle M \rangle$
   a. Cosa è successo a w?
   b. w è solo una stringa, come $\langle M \rangle$. Tutto ciò che stiamo facendo è definire quale stringa dare in input alla macchina.
3. D rifiuta $\langle D \rangle$ esattamente quando D accetta $\langle D \rangle$
   a. Questa è la contraddizione.
4. Dove si usa la diagonalizzazione?

Sommario

Un linguaggio non Turing-riconoscibile

- Abbiamo visto che A_{TM} è Turing-riconoscibile
- Sappiamo che l'insieme di tutte le TM è numerabile
- Sappiamo che l'insieme di tutti i linguaggi è non numerabile
- Di conseguenza deve esistere un linguaggio non Turing-riconiscibile

Un linguaggio non Turing-riconoscibile

- C'è ancora una cosa che dobbiamo fare prima di poter mostrare un linguaggio non Turing-riconoscibile.
- Mostreremo che se un linguaggio e il suo complemento sono Turing-riconoscibili, allora il linguaggio è decidibile.
- Un linguaggio è co-Turing riconoscibile se è il complemento di un linguaggio Turing-riconoscibile

Theorem

Un linguaggio è decidibile se solo se è Turing-riconoscibile e co-Turing riconoscibile.

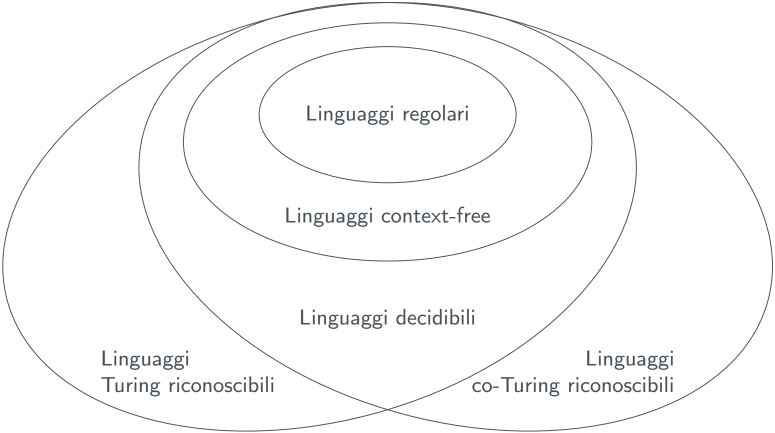
Dimostrazione:

- Dobbiamo dimostrare entrambe le direzioni
- Se A è decidibile, allora sia A che $\overline{A}$ sono Turing-riconoscibili
  - Il complementare di un linguaggio decidibile è decidibile!
- Se A e $\overline{A}$ sono Turing-riconoscibili, possiamo costruire un decisore per A

$\overline{A_{TM}}$ non è Turing-riconoscibile

- Se il complementare di A_{TM} fosse Turing-riconoscibile, allora A_{TM} sarebbe decidibile
- Sappiamo che A_{TM} non è decidibile, quindi il suo complementare non può essere Turing-riconoscibile!

Conclusione



Cosa c'é qui fuori?