ELEMENTI DI TEORIA DELLA COMPUTAZIONE

M.Anselmo

a.a. 2022/23

ACCETTO ... O NON ACCETTO?

9 maggio 2023

TEORIA DELLA COMPUTAZIONE

Obiettivo: analizzare i limiti della risoluzione dei "problemi" mediante "algoritmi".

Problemi = Linguaggi di stringhe

Algoritmi = Macchine di Turing

Proveremo che esistono problemi che possono essere risolti mediante algoritmi e altri no.

Problemi di decisione e linguaggi

I problemi di decisione sono problemi che hanno come soluzione una risposta SI o NO.
Il linguaggio associato a un problema di decisione è l'insieme delle codifiche di istanze del problema con risposta SI.

Esempio.

PRIMO: Dato un numero x, x è primo?

Il linguaggio associato al problema "PRIMO" è

$$L_{PRIMO} = \{\langle x \rangle \mid x \text{ è un numero primo}\}$$

dove $\langle x \rangle$ = "ragionevole" codifica di x mediante una stringa
Risolvere PRIMO equivale a decidere il linguaggio L_{PRIMO}.

In questo modo esprimiamo un problema computazionale come un problema di riconoscimento di un linguaggio.

Problemi indecidibili

Motivazioni per lo studio di questi problemi:

- Sapere che esistono problemi non risolvibili con un computer

Problemi indecidibili

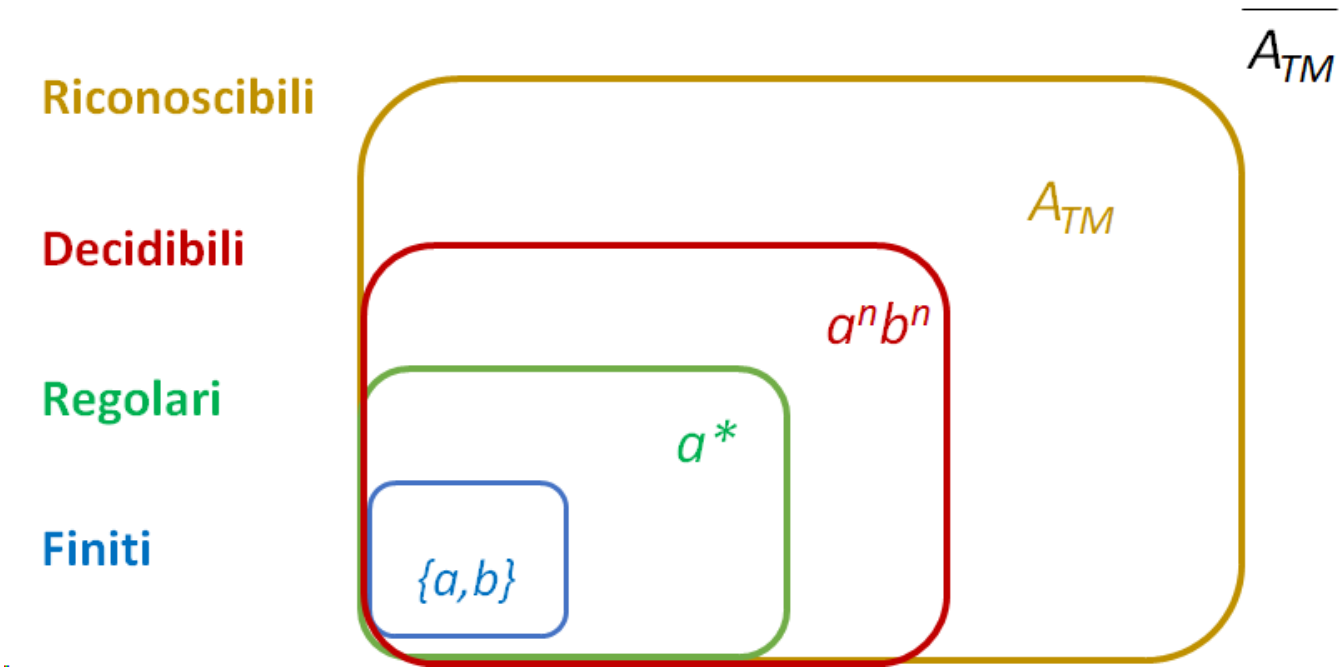
Motivazioni per lo studio di questi problemi:

- Sapere che esistono problemi non risolvibili con un computer

I problemi indecidibili sono esoterici o lontani dai problemi di interesse informatico? NO Esempi di problemi indecidibili:

- Il problema generale della verifica del software non è risolvibile mediante computer
  - Costruire un perfetto sistema di "debugging" per determinare se un programma si arresta.
  - Equivalenza di programmi: Dati due programmi essi forniscono lo stesso output?
- Compressione dati ottimale: Trovare il programma più corto per produrre una immagine data.
- Individuazione dei virus: Questo programma è un virus?

Risultati


Riconoscibili
Decidibili
Regolari
Finiti
$\overline{A_{TM}}$
A_{TM}
$a^n b^n$
$a*$
$\{a,b\}$

Strumenti

- Cardinalità di insiemi (infiniti)
- Metodo della diagonalizzazione di Cantor
- Autoreferenzialità

Cardinalità

Come si misura la cardinalità di insiemi infiniti?

Cantor osservò che due insiemi finiti hanno la stessa cardinalità se gli elementi dell'uno possono essere messi in corrispondenza uno a uno con quelli dell'altro.
Questo metodo confronta le dimensioni senza ricorrere al conteggio.

Estese questo concetto agli insiemi infiniti.

Cardinalità

Definizione

Due insiemi X e Y hanno la stessa cardinalità se esiste una funzione biettiva $f : X \to Y$ di X su Y.

$$|X| = |Y| \Leftrightarrow \text{ esiste una funzione biettiva } f : X \to Y$$

.

Esempio Sia $\mathbb{N}_P = \{2n \mid n \in \mathbb{N}\}$ l'insieme dei numeri naturali pari. La funzione $f : \mathbb{N} \to \mathbb{N}_P$ dove $f(n) = 2n$ è una funzione biettiva e quindi $\mathbb{N}_P$ e $\mathbb{N}$ hanno la stessa cardinalità, anche se $\mathbb{N}_P \subsetneq \mathbb{N}$.

n	$f(n)$
1	2
2	4
3	6
⋮	⋮

Insiemi numerabili

Definizione

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

Se A è numerabile possiamo "numerare" gli elementi di A e scrivere una lista $(a_1, a_2, \ldots)$

cioè per ogni numero naturale i, possiamo specificare l'elemento i–mo della lista.

Esempio L'insieme $\mathbb{N}_P$ dei numeri naturali pari è numerabile: l'elemento i-esimo della lista corrisponde a $2i$.

Insiemi numerabili: Σ^* e MdT

Esempio. L'insieme Σ^* di tutte le stringhe sull'alfabeto Σ è numerabile.

Possiamo elencare le stringhe secondo l'ordine radix, cioè per lunghezza e, a parità di lunghezza, in ordine lessicografico.
Per esempio: $\Sigma = \{0, 1\}$, $w_0 = \epsilon$, $w_1 = 0$, $w_2 = 1$, $w_3 = 00$, ...

Esempio. L'insieme

$$\{\langle M \rangle \mid M \text{ è una MdT sull'alfabeto } \Sigma\}$$

è numerabile.

Abbiamo visto che è possibile codificare le MdT tramite stringhe su un alfabeto (anche binario).
E l'insieme di tutte le stringhe su un alfabeto è numerabile.

Insiemi numerabili: Σ^* e MdT

Esempio. L'insieme Σ^* di tutte le stringhe sull'alfabeto Σ è numerabile.

Possiamo elencare le stringhe secondo l'ordine radix, cioè per lunghezza e, a parità di lunghezza, in ordine lessicografico.
Per esempio: $\Sigma = \{0, 1\}$, $w_0 = \epsilon$, $w_1 = 0$, $w_2 = 1$, $w_3 = 00$, ...

Esempio. L'insieme

$$\{\langle M \rangle \mid M \text{ è una MdT sull'alfabeto } \Sigma\}$$

è numerabile.

Abbiamo visto che è possibile codificare le MdT tramite stringhe su un alfabeto (anche binario).
E l'insieme di tutte le stringhe su un alfabeto è numerabile.

E' quindi numerabile anche l'insieme *RE* dei linguaggi riconosciuti da MdT (o Ricorsivamente Enumerabili): il primo sarà il linguaggio riconosciuto dalla prima MdT, il secondo dalla seconda, e così via.

L'insieme dei numeri reali non è numerabile

Teorema. L'insieme $\mathbb{R}$ dei numeri reali non è numerabile.

Lo dimostreremo col metodo della diagonalizzazione di Cantor.
Se per assurdo $\mathbb{R}$ fosse numerabile, allora potremmo elencare tutti i numeri reali:

$$f(1),\ f(2),\ f(3), \ldots$$

Per esempio:

n	$f(n)$
1	3.14159...
2	55.55555...
3	0.12345...
4	0.50000...
⋮	⋮

L'insieme dei numeri reali non è numerabile (cont.)

Concentriamoci sulle parti decimali e organizziamole in una matrice:

$i \backslash f(i)$	f_1	f_2	f_3	$\dots$	$\dots$	$\dots$
1	$f_1(1)$	$f_2(1)$	$f_3(1)$	$\dots$	$f_i(1)$	$\dots$
2	$f_1(2)$	$f_2(2)$	$f_3(2)$	$\dots$	$f_i(2)$	$\dots$
3	$f_1(3)$	$f_2(3)$	$f_3(3)$	$\dots$	$f_i(3)$	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
i	$f_1(i)$	$f_2(i)$	$f_3(i)$	$\dots$	$f_i(i)$	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Vedremo come costruire un numero $x \in \mathbb{R}$ che non è presente nell'elenco, col metodo della diagonale.

L'insieme dei numeri reali non è numerabile (cont.)

Per esempio:

n	$f(n)$
1	3.14159...
2	55.55555...
3	0.12345...
4	0.50000...
⋮	⋮

$i \backslash f(i)$	f_1	f_2	f_3	f_4	f_5	...
1	1	4	1	5	9	...
2	5	5	5	5	5	...
3	1	2	3	4	5	...
4	5	0	0	0	0	...
⋮	⋮	⋮	⋮	⋱	⋮	⋮
i	$f_1(i)$	$f_2(i)$	$f_3(i)$	...	$f_i(i)$	...
⋮	⋮	⋮	⋮	⋮	⋮	⋱

L'insieme dei numeri reali non è numerabile (cont.)

$i \backslash f(i)$	f_1	f_2	f_3	$\ldots$	$\ldots$	$\ldots$
1	$f_1(1)$	$f_2(1)$	$f_3(1)$	$\ldots$	$f_i(1)$	$\ldots$
2	$f_1(2)$	$f_2(2)$	$f_3(2)$	$\ldots$	$f_i(2)$	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
i	$f_1(i)$	$f_2(i)$	$f_3(i)$	$\ldots$	$f_i(i)$	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Sia $x \in (0,1)$ il numero $x = 0, x_1 x_2 \ldots x_i \ldots$
ottenuto scegliendo $x_i \neq f_i(i)$ per ogni $i \geq 1$. Chiaramente $x \in \mathbb{R}$.

L'insieme dei numeri reali non è numerabile (cont.)

$i \backslash f(i)$	f_1	f_2	f_3	$\ldots$	$\ldots$	$\ldots$
1	$f_1(1)$	$f_2(1)$	$f_3(1)$	$\ldots$	$f_i(1)$	$\ldots$
2	$f_1(2)$	$f_2(2)$	$f_3(2)$	$\ldots$	$f_i(2)$	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
i	$f_1(i)$	$f_2(i)$	$f_3(i)$	$\ldots$	$f_i(i)$	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Sia $x \in (0,1)$ il numero $x = 0, x_1 x_2 \ldots x_i \ldots$
ottenuto scegliendo $x_i \neq f_i(i)$ per ogni $i \geq 1$. Chiaramente $x \in \mathbb{R}$.

Il numero x compare nella lista?
Se $x = f(j)$ allora la sua j-esima cifra decimale soddisferebbe $x_j = f_j(j)$. Ma $x_j \neq f_j(j)$ (per def. di x): contraddizione!

Quindi $x \in \mathbb{R}$ non può comparire nella lista e $\mathbb{R}$ non è numerabile.

Buoni e cattivi

Numerabili	Non numerabili
$\mathbb{N}$	$\mathbb{R}$
$\mathbb{N}^2$	$\mathcal{B}$
$\mathbb{Q}_+$	$\mathcal{P}(\Sigma^*)$
Σ^*	
MdT	
RE	

$\mathcal{B}$ è l'insieme di tutte le sequenze binarie infinite.
E' possibile dimostrare che $\mathcal{B}$ non è numerabile col metodo di diagonalizzazione, in modo simile alla dimostrazione che $\mathbb{R}$ non è numerabile (esercizio).

$\mathcal{P}(\Sigma^*)$ non è numerabile

Teorema

L'insieme $\mathcal{P}(\Sigma^*)$ dei linguaggi su Σ non è numerabile.

E' possibile dimostrarlo in 2 modi.

- Dimostrazione 1 (come sul libro, via $\mathcal{B}$)
- Dimostrazione 2 (diagonalizzazione diretta)

Concludendo

Numerabili	Non numerabili
$\mathbb{N}$	$\mathbb{R}$
$\mathbb{N}^2$	$\mathcal{B}$
$\mathbb{Q}_+$	$\mathcal{P}(\Sigma^*)$
Σ^*	
MdT	
RE	

Esistono più linguaggi/problemi che macchine di Turing/algoritmi.

Esistono più linguaggi che linguaggi Turing riconoscibili.

Teorema

Esistono linguaggi che non sono Turing riconoscibili.

Linguaggi non riconoscibili

Teorema

Esistono linguaggi che non sono Turing riconoscibili.

Esistono.... ma abbiamo un esempio?

Linguaggi non riconoscibili

Teorema

Esistono linguaggi che non sono Turing riconoscibili.

Esistono.... ma abbiamo un esempio?

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

Nel seguito dimostreremo che:

- A_{TM} è riconoscibile
- A_{TM} non è decidibile
- $\overline{A_{TM}}$ non è riconoscibile.

Un problema indecidibile

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT e } M \text{ accetta } w\}$$

A_{TM} è il linguaggio associato al problema decisionale dell'accettazione di una macchina di Turing.

Teorema

Il linguaggio A_{TM} non è decidibile.

Un problema indecidibile

Dimostrazione.
Supponiamo per assurdo che esiste una macchina di Turing H con due possibili risultati di una computazione (accettazione, rifiuto), che decida il linguaggio A_{TM}. Il decisore H su input $\langle M, w\rangle$:

$$H = \begin{cases} \textit{accetta} & \text{se } \langle M, w\rangle \in A_{TM}, \text{ cioè se } M \text{ accetta } w \\ \textit{rifiuta} & \text{se } \langle M, w\rangle \notin A_{TM}, \text{ cioè se } M \text{ non accetta } w \end{cases}$$

$$\langle M, w\rangle \rightarrow \boxed{H} \rightarrow \begin{cases} \textit{accetta} & \text{se } M \text{ accetta } w \\ \textit{rifiuta} & \text{se } M \text{ non accetta } w \end{cases}$$

Un problema indecidibile

Costruiamo una nuova MdT D che usa H come sottoprogramma.
La MdT D sull'input $\langle M \rangle$, dove M è una MdT:

1. Simula H sull'input $\langle M, \langle M \rangle \rangle$
2. Fornisce come output l'opposto di H, cioè se H accetta, *rifiuta* e se H rifiuta, *accetta*

$$\langle M \rangle \rightarrow \boxed{\boxed{C} \rightarrow \langle M, \langle M \rangle \rangle \rightarrow \boxed{H} \rightarrow \begin{cases} \textit{accetta} \\ \textit{rifiuta} \end{cases} \rightarrow \boxed{I}} \rightarrow \begin{cases} \textit{rifiuta} \\ \textit{accetta} \end{cases}$$

Un problema indecidibile

Costruiamo una nuova MdT D che usa H come sottoprogramma.

$$\langle M \rangle \rightarrow \boxed{\boxed{C} \rightarrow \langle M, \langle M \rangle \rangle \rightarrow \boxed{H} \rightarrow \begin{cases} accetta \\ rifiuta \end{cases} \rightarrow \boxed{I}} \rightarrow \begin{cases} rifiuta \\ accetta \end{cases}$$

Un problema indecidibile

Costruiamo una nuova MdT D che usa H come sottoprogramma.

$$\langle M \rangle \rightarrow \boxed{\boxed{C} \rightarrow \langle M, \langle M \rangle \rangle \rightarrow \boxed{H} \rightarrow \begin{cases} accetta \\ rifiuta \end{cases} \rightarrow \boxed{I}} \rightarrow \begin{cases} rifiuta \\ accetta \end{cases}$$

Per costruzione:
D rifiuta $\langle M \rangle$ sse H accetta $\langle M, \langle M \rangle \rangle$ sse M accetta $\langle M \rangle$.

Un problema indecidibile

Costruiamo una nuova MdT D che usa H come sottoprogramma.

$$\langle M \rangle \to \boxed{\boxed{C} \to \langle M, \langle M \rangle \rangle \to \boxed{H} \to \begin{cases} accetta \\ rifiuta \end{cases} \to \boxed{I}} \to \begin{cases} rifiuta \\ accetta \end{cases}$$

Per costruzione:
D rifiuta $\langle M \rangle$ sse H accetta $\langle M, \langle M \rangle \rangle$ sse M accetta $\langle M \rangle$.

Quindi

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

Un problema indecidibile

Se ora diamo in input a D la sua stessa codifica $\langle D \rangle$ abbiamo

Un problema indecidibile

Se ora diamo in input a D la sua stessa codifica $\langle D \rangle$ abbiamo

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

$$\langle D \rangle \rightarrow \boxed{\boxed{C} \rightarrow \langle D, \langle D \rangle \rangle \rightarrow \boxed{H} \rightarrow \begin{cases} \textit{accetta} \\ \textit{rifiuta} \end{cases} \rightarrow \boxed{I}} \rightarrow \begin{cases} \textit{rifiuta} \\ \textit{accetta} \end{cases}$$

Un problema indecidibile

Se ora diamo in input a D la sua stessa codifica $\langle D \rangle$ abbiamo

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

$$\langle D \rangle \rightarrow \boxed{\boxed{C} \rightarrow \langle D, \langle D \rangle \rangle \rightarrow \boxed{H} \rightarrow \begin{cases} \textit{accetta} \\ \textit{rifiuta} \end{cases} \rightarrow \boxed{I}} \rightarrow \begin{cases} \textit{rifiuta} \\ \textit{accetta} \end{cases}$$

Cioè D accetta $\langle D \rangle$ se e solo se D non accetta $\langle D \rangle$.

Assurdo!

Tutto causato dall'assunzione che esiste H. Quindi H non esiste!

□

A_{TM} è indecidibile: riepilogo della dimostrazione

1. Definiamo $A_{TM} = \{\langle M, w\rangle \mid M \text{ è MdT che accetta } w\}$
2. Assumiamo A_{TM} decidibile; sia H MdT che lo decide
3. Usiamo H per costruire MdT D che inverte le decisioni; $D(\langle M\rangle)$: accetta se M non accetta $\langle M\rangle$; rifiuta se M accetta $\langle M\rangle$.
4. Diamo in input a D la sua codifica $\langle D\rangle$: $D(\langle D\rangle)$ accetta sse D rifiuta.

Contraddizione

Diagonalizzazione?

Dove è intervenuto il metodo della diagonalizzazione in questa prova?
Consideriamo la tavola dove il coefficiente (i, j) è acc se M_i accetta $\langle M_j \rangle$.

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\ldots$
M_1	acc		acc		$\ldots$
M_2	acc	acc	acc	acc	$\ldots$
M_3					
M_4	acc	acc			$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Diagonalizzazione?

Consideriamo H, il presunto decisore di A_{TM}, e la tavola dove il coefficiente (i, j) è il valore di H su $\langle M_i, \langle M_j \rangle \rangle$.
Essendo un decisore, H rifiuta anche se M_i con input $\langle M_j \rangle$ va in loop (oltre a se M_i rifiuta).

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\ldots$
M_1	acc	rej	acc	rej	$\ldots$
M_2	acc	acc	acc	acc	$\ldots$
M_3	rej	rej	rej	rej	$\ldots$
M_4	acc	acc	rej	rej	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Diagonalizzazione?

Se D esistesse, ci sarebbe contraddizione in corrispondenza di $(\langle D \rangle)$. Ricorda che per costruzione:
D rifiuta $\langle M \rangle$ sse H accetta $\langle M, \langle M \rangle \rangle$ sse M accetta $\langle M \rangle$.

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\ldots$	$\langle D \rangle$	$\ldots$
M_1	acc	rej	acc	rej	$\ldots$	$\ldots$	$\ldots$
M_2	acc	acc	acc	acc	$\ldots$	$\ldots$	$\ldots$
M_3	rej	rej	rej	rej	$\ldots$	$\ldots$	$\ldots$
M_4	acc	acc	rej	rej	$\ldots$	$\ldots$	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
D	acc	acc	rej	rej	$\ldots$	???	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Risultati

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

- A_{TM} non è decidibile OK
- A_{TM} è riconoscibile
- $\overline{A_{TM}}$ non è riconoscibile.

Risultati

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

- A_{TM} non è decidibile OK
- A_{TM} è riconoscibile
- $\overline{A_{TM}}$ non è riconoscibile.

Per dimostrare che A_{TM} è riconoscibile costruiamo una MdT speciale, che ha un interesse oltre questa dimostrazione.

Riconoscere A_{TM}

- Una MdT U che riconosce
  $A_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT che accetta la parola } w\}$
  riceve in input una **codifica** $\langle M, w\rangle$ di M e di un possibile input w di M e la accetta se M accetta w.

Riconoscere A_{TM}

- Una MdT U che riconosce
  $A_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT che accetta la parola } w\}$
  riceve in input una **codifica** $\langle M, w\rangle$ di M e di un possibile input w di M e la accetta se M accetta w.
- Più precisamente costruiremo U in modo che

$$\langle M, w\rangle \rightarrow \boxed{U} \rightarrow \begin{cases} \textit{accetta} & \text{se } M \text{ accetta } w \\ \textit{rifiuta} & \text{se } M \text{ rifiuta } w \\ \textit{non termina} & \text{se } M \text{ non termina} \end{cases}$$

Riconoscere A_{TM}

- Una MdT U che riconosce
  $A_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT che accetta la parola } w\}$
  riceve in input una **codifica** $\langle M, w\rangle$ di M e di un possibile input w di M e la accetta se M accetta w.
- Più precisamente costruiremo U in modo che

$$\langle M, w\rangle \rightarrow \boxed{U} \rightarrow \begin{cases} \textit{accetta} & \text{se } M \text{ accetta } w \\ \textit{rifiuta} & \text{se } M \text{ rifiuta } w \\ \textit{non termina} & \text{se } M \text{ non termina} \end{cases}$$

- In altre parole, la MdT U simula la computazione di una qualsiasi MdT M su un qualsiasi input w.

Macchina di Turing Universale

La MdT U è detta MdT universale perché simula la computazione di una qualsiasi MdT M su un qualsiasi input w. E' un esempio della MdT universale introdotta da A. Turing nel 1936.

Macchina di Turing Universale

La MdT U è detta MdT universale perché simula la computazione di una qualsiasi MdT M su un qualsiasi input w. E' un esempio della MdT universale introdotta da A. Turing nel 1936.

Anticipò alcuni sviluppi fondamentali in informatica:
- Compilatore C (risp. C++, Java) scritto in C (risp. in C++, in Java)
- Computer a programma memorizzato: computer che riesce a simulare/eseguire qualsiasi programma su qualsiasi input.

Macchina di Turing Universale

Dunque, la macchina di Turing Universale può essere pensata come il progetto logico di un calcolatore che riceve la codifica di un programma (M) e di dati in ingresso (w) ed esegue le istruzioni.

Macchina di Turing Universale

Dunque, la macchina di Turing Universale può essere pensata come il progetto logico di un calcolatore che riceve la codifica di un programma (M) e di dati in ingresso (w) ed esegue le istruzioni.

Al primo anno abbiamo visto uno schema semplificato di un processore MIPS che avendo in memoria una stringa binaria che codifica una sequenza di istruzioni (M) e una stringa binaria (a 32 bit) in ingresso (w) esegue le istruzioni e aggiorna i dati.
Faremo qualcosa di simile con una MdT e la sua codifica binaria come sequenza di transizioni.

A_{TM} è Turing riconoscibile

Teorema

Il linguaggio

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

è Turing riconoscibile.

Dimostrazione

La seguente macchina di Turing U riconosce A_{TM}.

U = "Sull'input $\langle M, w \rangle$ dove M è una TM e w è una stringa

1. Simula M sull'input w.
2. Se M accetta w, accetta l'input $\langle M, w \rangle$; se M rifiuta w, rifiuta l'input $\langle M, w \rangle$."

U rifiuta ogni stringa che non sia della forma $\langle M, w \rangle$ dove M è una TM e w è una stringa.

Quindi U accetta una stringa y se e solo se y è della forma $\langle M, w \rangle$ dove M è una TM, w è una stringa e M accetta w.

In altri termini, U accetta una stringa y se e solo se $y = \langle M, w \rangle$ è un elemento di A_{TM}.

Ne segue $L(U) = A_{TM}$.

Macchina di Turing Universale

Ma com'è fatta la macchina di Turing Universale?

Possiamo pensare a una macchina di Turing U a tre nastri.

La macchina U riceve in input la codifica $\langle M, w \rangle$ di M e w.

Prima di "eseguire" M su w, U esegue alcuni passi di inizializzazione:

1. copia sul secondo nastro la codifica di M,
2. copia sul terzo nastro la codifica dello stato iniziale di M,
3. lascia sul primo nastro la codifica di w.

Macchina di Turing Universale

Durante la sua computazione U:

- usa il primo nastro per simulare la computazione di M,
- lascia sul secondo nastro la codifica di M,
- ha sul terzo nastro la codifica dello stato corrente di M.

Macchina di Turing Universale

Durante la sua computazione U:

- usa il primo nastro per simulare la computazione di M,
- lascia sul secondo nastro la codifica di M,
- ha sul terzo nastro la codifica dello stato corrente di M.

U individua l'istruzione corrente sul secondo nastro, usando il contenuto del terzo nastro e il simbolo corrente (codificato) sul primo nastro, quindi decodifica l'istruzione e la esegue.

Note

1. **Nota:** U è detta MdT universale.
   E' un esempio della MdT universale introdotta da A. Turing nel 1936. Anticipò alcuni sviluppi fondamentali in informatica:
   - Compilatore C (risp. C++, Java) scritto in C (risp. in C++, in Java)
   - Computer a programma memorizzato: computer che riesce a simulare/eseguire qualsiasi programma su qualsiasi input.
   Dunque, la macchina di Turing Universale può essere pensata come il progetto logico di un calcolatore.

2. **Nota:** U riconosce A_{TM}: accetta ogni coppia $\langle M, w \rangle \in A_{TM}$.

Note

1. **Nota:** U è detta MdT universale.
   E' un esempio della MdT universale introdotta da A. Turing nel 1936. Anticipò alcuni sviluppi fondamentali in informatica:
   - Compilatore C (risp. C++, Java) scritto in C (risp. in C++, in Java)
   - Computer a programma memorizzato: computer che riesce a simulare/eseguire qualsiasi programma su qualsiasi input.
   Dunque, la macchina di Turing Universale può essere pensata come il progetto logico di un calcolatore.

2. **Nota:** U riconosce A_{TM}: accetta ogni coppia $\langle M, w\rangle \in A_{TM}$.

3. **Nota:** U cicla su $\langle M, w\rangle$ se (e solo se) M cicla su w. Quindi U **non decide** A_{TM}.

Risultati

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

- A_{TM} non è decidibile OK
- A_{TM} è riconoscibile OK
- $\overline{A_{TM}}$ non è riconoscibile.

Risultati

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

- A_{TM} non è decidibile OK
- A_{TM} è riconoscibile OK
- $\overline{A_{TM}}$ non è riconoscibile.

Utilizzando il metodo della diagonalizzazione, abbiamo provato che esistono linguaggi che non sono Turing riconoscibili. Ma ancora non abbiamo visto un esempio di un tale linguaggio.
Per dimostrare che $\overline{A_{TM}}$ non è riconoscibile premettiamo alcuni risultati necessari.

Una proprietà dei linguaggi decidibili

Definizione

Diciamo che un linguaggio L è co-Turing riconoscibile se $\overline{L}$ è Turing riconoscibile.

Una proprietà dei linguaggi decidibili

Definizione

Diciamo che un linguaggio L è co-Turing riconoscibile se $\overline{L}$ è Turing riconoscibile.

Teorema

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

Una proprietà dei linguaggi decidibili

L è decidibile $\Leftrightarrow$ L e il suo complemento sono entrambi Turing riconoscibili.

Una proprietà dei linguaggi decidibili

L è decidibile $\Leftrightarrow$ L e il suo complemento sono entrambi Turing riconoscibili.

Dimostrazione

($\Rightarrow$) Se L è decidibile allora esiste una macchina di Turing M con due possibili risultati di una computazione (accettazione, rifiuto) e tale che M accetta w se e solo se $w \in L$. Allora L è Turing riconoscibile. Inoltre è facile costruire una MdT $\overline{M}$ che accetta w se e solo se $w \notin L$:

Una proprietà dei linguaggi decidibili

L è decidibile $\Leftrightarrow$ L e il suo complemento sono entrambi Turing riconoscibili.

Dimostrazione

($\Rightarrow$) Se L è decidibile allora esiste una macchina di Turing M con due possibili risultati di una computazione (accettazione, rifiuto) e tale che M accetta w se e solo se $w \in L$. Allora L è Turing riconoscibile. Inoltre è facile costruire una MdT $\overline{M}$ che accetta w se e solo se $w \notin L$:

$$w \rightarrow \boxed{\boxed{M} \rightarrow \begin{cases} \textit{accetta} & \text{se } w \in L \\ \textit{rifiuta} & \text{se } w \notin L \end{cases}} \rightarrow \begin{cases} \textit{rifiuta} & \text{se } M \text{ accetta} \\ \textit{accetta} & \text{se } M \text{ rifiuta} \end{cases}$$

Una proprietà dei linguaggi decidibili

($\Leftarrow$) Supponiamo che L e il suo complemento siano entrambi Turing riconoscibili. Sia M_1 una MdT che riconosce L e M_2 una MdT che riconosce $\overline{L}$.

Una proprietà dei linguaggi decidibili

($\Leftarrow$) Supponiamo che L e il suo complemento siano entrambi Turing riconoscibili. Sia M_1 una MdT che riconosce L e M_2 una MdT che riconosce $\overline{L}$.

Definiamo una MdT N (a due nastri): sull'input x

1. Copia x sui nastri di M_1 e M_2

Una proprietà dei linguaggi decidibili

($\Leftarrow$) Supponiamo che L e il suo complemento siano entrambi Turing riconoscibili. Sia M_1 una MdT che riconosce L e M_2 una MdT che riconosce $\overline{L}$.

Definiamo una MdT N (a due nastri): sull'input x

1. Copia x sui nastri di M_1 e M_2
2. Simula M_1 e M_2 in parallelo (usa un nastro per M_1, l'altro per M_2, alternando un passo di M_1 con uno di M_2)

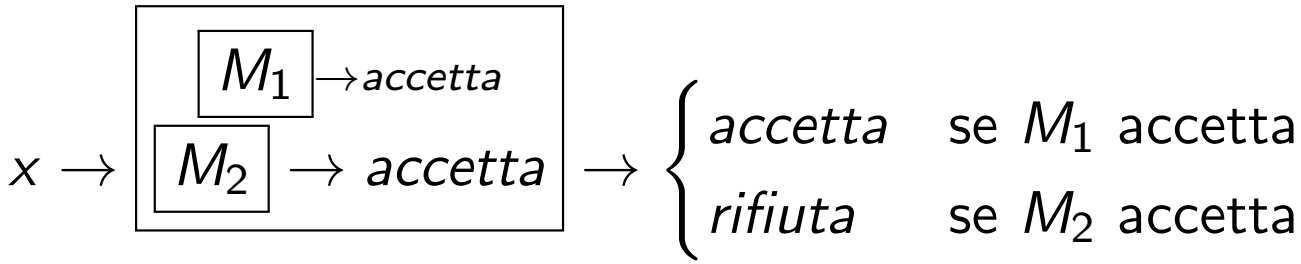
Una proprietà dei linguaggi decidibili

($\Leftarrow$) Supponiamo che L e il suo complemento siano entrambi Turing riconoscibili. Sia M_1 una MdT che riconosce L e M_2 una MdT che riconosce $\overline{L}$.

Definiamo una MdT N (a due nastri): sull'input x

1. Copia x sui nastri di M_1 e M_2
2. Simula M_1 e M_2 in parallelo (usa un nastro per M_1, l'altro per M_2, alternando un passo di M_1 con uno di M_2)
3. Se M_1 accetta, *accetta*; se M_2 accetta, *rifiuta*

Una proprietà dei linguaggi decidibili

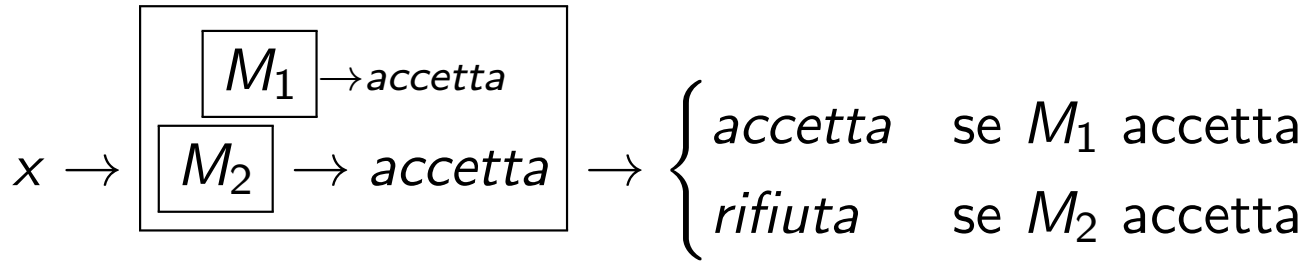


Una proprietà dei linguaggi decidibili

$$x \to \boxed{\begin{array}{l} \boxed{M_1} \to \textit{accetta} \\ \boxed{M_2} \to \textit{accetta} \end{array}} \to \begin{cases} \textit{accetta} & \text{se } M_1 \text{ accetta} \\ \textit{rifiuta} & \text{se } M_2 \text{ accetta} \end{cases}$$

N decide L. Infatti, per ogni stringa x abbiamo due casi:

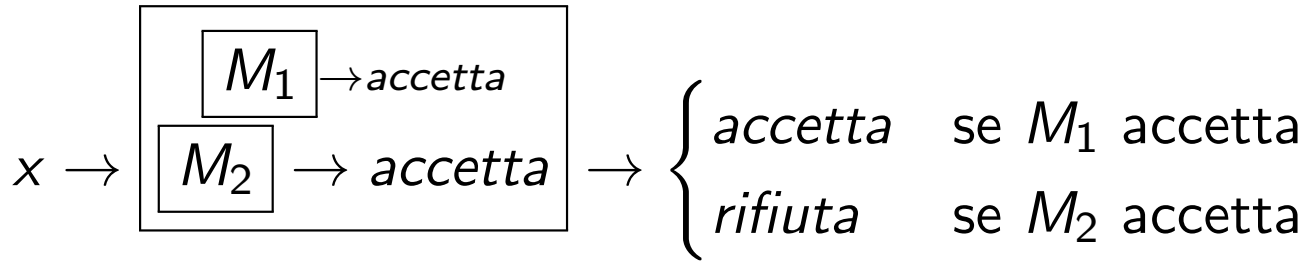
Una proprietà dei linguaggi decidibili



N decide L. Infatti, per ogni stringa x abbiamo due casi:

1. $x \in L$. Ma $x \in L$ se e solo se M_1 si arresta e accetta x. Quindi N accetta x.

Una proprietà dei linguaggi decidibili



N decide L. Infatti, per ogni stringa x abbiamo due casi:

1. $x \in L$. Ma $x \in L$ se e solo se M_1 si arresta e accetta x. Quindi N accetta x.
2. $x \notin L$. Ma $x \notin L$ se e solo se M_2 si arresta e accetta x. Quindi N rifiuta x.

Una proprietà dei linguaggi decidibili

$$x \rightarrow \boxed{\begin{array}{l} \boxed{M_1} \rightarrow accetta \\ \boxed{M_2} \rightarrow accetta \end{array}} \rightarrow \begin{cases} accetta & \text{se } M_1 \text{ accetta} \\ rifiuta & \text{se } M_2 \text{ accetta} \end{cases}$$

N decide L. Infatti, per ogni stringa x abbiamo due casi:

1. $x \in L$. Ma $x \in L$ se e solo se M_1 si arresta e accetta x. Quindi N accetta x.
2. $x \notin L$. Ma $x \notin L$ se e solo se M_2 si arresta e accetta x. Quindi N rifiuta x.

Poichè una e solo una delle due MdT accetta x, N è una MdT con solo due possibili risultati di una computazione (accettazione, rifiuto) e tale che N accetta x se e solo se $x \in L$.

□

Un linguaggio che non è Turing riconoscibile

Teorema
$\overline{A_{TM}}$ non è Turing riconoscibile.

Dimostrazione.

□

Un linguaggio che non è Turing riconoscibile

Teorema

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

Dimostrazione.

Supponiamo per assurdo che $\overline{A_{TM}}$ sia Turing riconoscibile.

□

Un linguaggio che non è Turing riconoscibile

Teorema

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

Dimostrazione.

Supponiamo per assurdo che $\overline{A_{TM}}$ sia Turing riconoscibile.
Sappiamo che A_{TM} è Turing riconoscibile.

Un linguaggio che non è Turing riconoscibile

Teorema

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

Dimostrazione.

Supponiamo per assurdo che $\overline{A_{TM}}$ sia Turing riconoscibile.
Sappiamo che A_{TM} è Turing riconoscibile.
Quindi A_{TM} è Turing riconoscibile e co-Turing riconoscibile.

□

Un linguaggio che non è Turing riconoscibile

Teorema

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

Dimostrazione.

Supponiamo per assurdo che $\overline{A_{TM}}$ sia Turing riconoscibile.
Sappiamo che A_{TM} è Turing riconoscibile.
Quindi A_{TM} è Turing riconoscibile e co-Turing riconoscibile.
Per il precedente teorema, A_{TM} è decidibile.

□

Un linguaggio che non è Turing riconoscibile

Teorema

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

Dimostrazione.

Supponiamo per assurdo che $\overline{A_{TM}}$ sia Turing riconoscibile.
Sappiamo che A_{TM} è Turing riconoscibile.
Quindi A_{TM} è Turing riconoscibile e co-Turing riconoscibile.
Per il precedente teorema, A_{TM} è decidibile.
Assurdo, poichè abbiamo dimostrato che A_{TM} è indecidibile.

□

Esercizio 1

La classe dei linguaggi **decidibili** è chiusa rispetto al complemento?

Esercizio 1

La classe dei linguaggi **decidibili** è chiusa rispetto al complemento?

Soluzione:

La classe dei linguaggi decidibili è chiusa rispetto al complemento. Sia A un linguaggio decidibile, sia M_A una macchina di Turing che decide A.

Definiamo la macchina di Turing $M_{\overline{A}}$: sull'input w, $M_{\overline{A}}$ simula M_A e accetta w se e solo se M_A rifiuta w.

Poiché M_A si arresta su ogni input anche $M_{\overline{A}}$ si arresta su ogni input.

Inoltre, il linguaggio di $M_{\overline{A}}$ è $\overline{A}$ perché $M_{\overline{A}}$ accetta w se e solo se M_A rifiuta w e quindi se e solo se $w \notin A$.

Quindi $M_{\overline{A}}$ è una macchina di Turing che decide $\overline{A}$ e $\overline{A}$ è decidibile.

Soluzione formale

Formalmente, se

$$M_A = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$$

definiamo

$$M_{\overline{A}} = (Q, \Sigma, \Gamma, \delta', q_0, q_{accept}, q_{reject})$$

dove, per ogni $q \in Q \setminus \{q_{accept}, q_{reject}\}$, per ogni $\gamma \in \Gamma$

$$\delta'(q,\gamma) = \begin{cases} \delta(q,\gamma) & \text{se } \delta(q,\gamma) = (q',\gamma',d), \\ & \text{con } q' \notin \{q_{accept}, q_{reject}\}, \\ (q_{accept},\gamma',d) & \text{se } \delta(q,\gamma) = (q_{reject},\gamma',d), \\ (q_{reject},\gamma',d) & \text{se } \delta(q,\gamma) = (q_{accept},\gamma',d) \end{cases}$$

Esercizio 2

La classe dei linguaggi **riconoscibili** è chiusa rispetto al complemento?

Esercizio 3

$\mathcal{B}$ è l'insieme di tutte le sequenze binarie infinite.
Dimostrare che $\mathcal{B}$ non è numerabile col metodo di diagonalizzazione, in modo simile alla dimostrazione che $\mathbb{R}$ non è numerabile.