DECIDIBILITÀ E INDECIDIBILITÀ

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

Studieremo: il potere computazionale degli algoritmi nella soluzione dei problemi.

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

Ricorda: Problemi di decisione

I problemi di decisione sono problemi che hanno come soluzione una risposta SI o NO.
Rappresenteremo i problemi di decisione mediante linguaggi.
Esempio. PRIMO: Dato un numero x, x è primo?
Il linguaggio che rappresenta "PRIMO" è

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

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

In questo modo esprimiamo un problema computazionale come un problema di riconoscimento di un linguaggio (insieme delle codifiche di istanze SI per il problema).

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?

Problemi indecidibili

OBIETTIVO: presentare un problema irrisolvibile

Problemi indecidibili

OBIETTIVO: presentare un problema irrisolvibile

Problema
Input: MdT M, stringa w
Domanda: M si arresta su input w?

Linguaggio corrispondente

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT che si arresta su } w\}$$

Problemi indecidibili

OBIETTIVO: presentare un problema irrisolvibile

Problema
Input: MdT M, stringa w
Domanda: M si arresta su input w?

Linguaggio corrispondente

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT che si arresta su } w\}$$

Teorema

Il linguaggio $HALT_{TM}$ non è decidibile.

Strumenti

Cardinalità di insiemi (infiniti).

Diagonalizzazione: metodo introdotto da Cantor.

Cardinalità

Cardinalità di un insieme: la sua taglia

Due insiemi hanno la stessa cardinalità se è possibile stabilire una corrispondenza tra i loro elementi.

Es: $A = \{1, 2, 3\}, B = \{4, 3, 5\} \qquad \Rightarrow \qquad 1 - 4, 2 - 3, 3 - 5$

Cardinalità

Cardinalità di un insieme: la sua taglia

Due insiemi hanno la stessa cardinalità se è possibile stabilire una corrispondenza tra i loro elementi.

Es: $A = \{1, 2, 3\}, B = \{4, 3, 5\} \qquad \Rightarrow \qquad 1 - 4, 2 - 3, 3 - 5$

Cosa possiamo dire per insiemi infiniti?

Paradosso di Hilbert

Nel paese senza confini esiste il più grande di tutti gli alberghi: l'albergo con infinite stanze. Tuttavia anche gli ospiti sono infiniti, e il proprietario ha esposto un cartello con la scritta COMPLETO.

Paradosso di Hilbert

Nel paese senza confini esiste il più grande di tutti gli alberghi: l'albergo con infinite stanze. Tuttavia anche gli ospiti sono infiniti, e il proprietario ha esposto un cartello con la scritta COMPLETO.
Ad un tratto si presenta un viaggiatore che ha assolutamente bisogno di una camera per la notte. Egli non fa questione di prezzo e infine convince l'albergatore, il quale trova il modo di alloggiarlo.
Come fa?

Paradosso di Hilbert

Nel paese senza confini esiste il più grande di tutti gli alberghi: l'albergo con infinite stanze. Tuttavia anche gli ospiti sono infiniti, e il proprietario ha esposto un cartello con la scritta COMPLETO.
Ad un tratto si presenta un viaggiatore che ha assolutamente bisogno di una camera per la notte. Egli non fa questione di prezzo e infine convince l'albergatore, il quale trova il modo di alloggiarlo.
Come fa?
Sposta tutti i clienti nella camera successiva (l'ospite della 1 alla 2, quello della 2 alla 3, etc.); in questo modo, è possibile, essendo l'albergo infinito, sistemare (nella camera 1) il nuovo ospite anche se l'albergo è pieno.

Paradosso di Hilbert-2

Poco dopo arriva una comitiva di **infiniti turisti**, anche in questo caso l'albergatore si lascia convincere, in fondo si tratta di un grosso affare, e trova posto ai **nuovi infiniti ospiti** con la stessa facilit con cui aveva alloggiato l'ospite in più
Come fa (senza ripetere infinite volte il passo visto prima)?

Paradosso di Hilbert-2

Poco dopo arriva una comitiva di **infiniti turisti**, anche in questo caso l'albergatore si lascia convincere, in fondo si tratta di un grosso affare, e trova posto ai **nuovi infiniti ospiti** con la stessa facilit con cui aveva alloggiato l'ospite in più

Come fa (senza ripetere infinite volte il passo visto prima)?

Sposta ogni ospite nella stanza con numero doppio rispetto a quello attuale (dalla 1 alla 2, dalla 2 alla 4,etc.), lasciando ai nuovi ospiti tutte le camere con i numeri dispari, che sono essi stessi infiniti, risolvendo dunque il problema.

Gli ospiti sono tutti dunque sistemati, benchè l'albergo fosse pieno.

Paradosso di Hilbert-3

Ancora piú difficile: ci sono infiniti alberghi con infinite stanze tutti al completo. Tutti gli alberghi chiudono, tranne uno. Tutti gli ospiti vogliono alloggiare nell'unico albergo rimasto aperto.

Come fa (senza ripetere infinite volte il passo visto prima)?

Paradosso di Hilbert-3

Ancora piú difficile: ci sono infiniti alberghi con infinite stanze tutti al completo. Tutti gli alberghi chiudono, tranne uno. Tutti gli ospiti vogliono alloggiare nell'unico albergo rimasto aperto.

Come fa (senza ripetere infinite volte il passo visto prima)?

Assegna ad ogni persona una coppia di numeri (n, m) in cui n indica l'albergo di provenienza, e m la relativa stanza. Gli ospiti sono quindi etichettati in questo modo:

$$\begin{array}{ccccc} (1,1) & (1,2) & (1,3) & (1,4) & \dots \\ (2,1) & (2,2) & (2,3) & (2,4) & \dots \\ (3,1) & (3,2) & (3,3) & (3,4) & \dots \\ (4,1) & (4,2) & (4,3) & (4,4) & \dots \\ \dots & \dots & \dots & \dots & \dots \end{array}$$

A questo punto basta assegnare le nuove stanze agli ospiti secondo un criterio ordinato, ad esempio per diagonali:

$(1,1) \to 1$; $(1,2) \to 2$; $(2,1) \to 3$; $(1,3) \to 4$; $(2,2) \to 5$; $(3,1) \to 6$; ...

Cardinalità

Quanti numeri naturali ci sono? INFINITI!

Quanti numeri naturali pari ci sono? INFINITI!

Quanti numeri naturali dispari ci sono? INFINITI!

Quanti numeri **reali** ci sono? INFINITI!

Cardinalità

Quanti numeri naturali ci sono? INFINITI!
Quanti numeri naturali pari ci sono? INFINITI!
Quanti numeri naturali dispari ci sono? INFINITI!

Quanti numeri **reali** ci sono? INFINITI!

La quantità di numeri reali è la stessa di quella dei numeri naturali?

Come si misura la cardinalità di insiemi infiniti?

Il Metodo della diagonalizzazione

Introdotto da Cantor nel 1973 mentre cercava di determinare come stabilire se, dati due insiemi infiniti, uno è più grande dell'altro.

Il Metodo della diagonalizzazione

Introdotto da Cantor nel 1973 mentre cercava di determinare come stabilire se, dati due insiemi infiniti, uno è più grande dell'altro.

- 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.

Il Metodo della diagonalizzazione

Introdotto da Cantor nel 1973 mentre cercava di determinare come stabilire se, dati due insiemi infiniti, uno è più grande dell'altro.

- 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.
- Estese questo concetto agli insiemi infiniti.

Premessa: Funzioni

Definizione

Una funzione $f : X \to Y$ è una relazione input-output.
*X è l'insieme dei possibili input (**dominio**),*
*Y è l'insieme dei possibili output (**codominio**).*
Per ogni input $x \in X$ esiste un solo output $y = f(x) \in Y$.

Premessa: Funzioni

Definizione

Una funzione $f : X \to Y$ *è una relazione input-output.*
X *è l'insieme dei possibili input (***dominio***),*
Y *è l'insieme dei possibili output (***codominio***).*
Per ogni input $x \in X$ *esiste un solo output* $y = f(x) \in Y$.

Definizione

$f : X \to Y$ *è iniettiva se* $\forall x, x' \in X \quad x \neq x' \Rightarrow f(x) \neq f(x')$

Premessa: Funzioni

Definizione

Una funzione $f : X \to Y$ *è una relazione input-output.*
X *è l'insieme dei possibili input (***dominio***),*
Y *è l'insieme dei possibili output (***codominio***).*
Per ogni input $x \in X$ *esiste un solo output* $y = f(x) \in Y$*.*

Definizione

$f : X \to Y$ *è iniettiva se* $\forall x, x' \in X \quad x \neq x' \Rightarrow f(x) \neq f(x')$

Definizione

$f : X \to Y$ *è suriettiva se* $\forall y \in Y$ *si ha* $y = f(x)$ *per qualche* $x \in X$

Premessa: Funzioni

Definizione

Una funzione $f : X \to Y$ *è una relazione input-output.*
X *è l'insieme dei possibili input (***dominio***),*
Y *è l'insieme dei possibili output (***codominio***).*
Per ogni input $x \in X$ *esiste un solo output* $y = f(x) \in Y$.

Definizione

$f : X \to Y$ *è iniettiva se* $\forall x, x' \in X \quad x \neq x' \Rightarrow f(x) \neq f(x')$

Definizione

$f : X \to Y$ *è suriettiva se* $\forall y \in Y$ *si ha* $y = f(x)$ *per qualche* $x \in X$

Definizione

Una funzione $f : X \to Y$ *è una funzione biettiva di* X *su* Y *(o una biezione tra* X *e* Y*) se* f *è iniettiva e suriettiva.*

Premessa: Funzioni

Esempio $f : \{1, 2, 5\} \to \{2, 4, 7\}$
$1 \to 2$, $2 \to 2$, $5 \to 4$ è una funzione. Non è nè iniettiva nè suriettiva.

Premessa: Funzioni

Esempio $f : \{1, 2, 5\} \to \{2, 4, 7\}$
$1 \to 2$, $2 \to 2$, $5 \to 4$ è una funzione. Non è nè iniettiva nè suriettiva.

Esempio $f : \{1, 2, 5\} \to \{2, 4, 7, 9\}$
$1 \to 2$, $2 \to 4$, $5 \to 7$ è una funzione iniettiva ma non suriettiva.

Premessa: Funzioni

Esempio $f : \{1, 2, 5\} \to \{2, 4, 7\}$
$1 \to 2$, $2 \to 2$, $5 \to 4$ è una funzione. Non è nè iniettiva nè suriettiva.

Esempio $f : \{1, 2, 5\} \to \{2, 4, 7, 9\}$
$1 \to 2$, $2 \to 4$, $5 \to 7$ è una funzione iniettiva ma non suriettiva.

Esempio $f : \{1, 2, 5\} \to \{2, 4\}$
$1 \to 2$, $2 \to 4$, $5 \to 2$ è una funzione suriettiva ma non iniettiva.

Premessa: Funzioni

Esempio $f : \{1, 2, 5\} \to \{2, 4, 7\}$
$1 \to 2$, $2 \to 2$, $5 \to 4$ è una funzione. Non è nè iniettiva nè suriettiva.

Esempio $f : \{1, 2, 5\} \to \{2, 4, 7, 9\}$
$1 \to 2$, $2 \to 4$, $5 \to 7$ è una funzione iniettiva ma non suriettiva.

Esempio $f : \{1, 2, 5\} \to \{2, 4\}$
$1 \to 2$, $2 \to 4$, $5 \to 2$ è una funzione suriettiva ma non iniettiva.

Esempio $f : \{1, 2, 5\} \to \{2, 4, 7\}$
$1 \to 2$, $2 \to 4$, $5 \to 7$ è una funzione biettiva.

Premessa: Cardinalità

Definizione

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

Premessa: Cardinalità

Definizione

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

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

Premessa: 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 $f : \{1, 2, 5\} \to \{2, 4, 7\}$ $\qquad$ $1 \to 2$, $2 \to 4$, $5 \to 7$.

Premessa: 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 $f : \{1, 2, 5\} \to \{2, 4, 7\}$ $\qquad 1 \to 2,\ 2 \to 4,\ 5 \to 7.$

Esempio $f : \mathbb{N} \to \{2n \mid n \in \mathbb{N}\}$ $\qquad n \to 2n$

Insiemi numerabili

Definizione

Un insieme è enumerabile (o numerabile) se ha la stessa cardinalità di un sottoinsieme di $\mathbb{N}$.

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

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

Es. Per l'insieme dei numeri naturali pari, l'elemento i-mo della lista corrisponde a $2i$.

Insiemi numerabili

Es. L'insieme dei numeri razionali è numerabile

Mostriamo che possiamo formare una lista di tutti i numeri razionali
Formiamo un rettangolo infinito:

$$\begin{array}{ccccc}
1/1 & 1/2 & 1/3 & 1/4 & \dots \\
2/1 & 2/2 & 2/3 & 2/4 & \dots \\
3/1 & 3/2 & 3/3 & 3/4 & \dots \\
4/1 & 4/2 & 4/3 & 4/4 & \dots \\
\dots & \dots & \dots & \dots & \dots
\end{array}$$

Insiemi numerabili

Come formiamo la lista di tutti i numeri razionali?
Se prendiamo prima tutta la prima linea, non arriviamo mai alla seconda!

$$\begin{array}{ccccc}
1/1 & 1/2 & 1/3 & 1/4 & \dots \\
2/1 & 2/2 & 2/3 & 2/4 & \dots \\
3/1 & 3/2 & 3/3 & 3/4 & \dots \\
4/1 & 4/2 & 4/3 & 4/4 & \dots \\
\dots & \dots & \dots & \dots & \dots
\end{array}$$

Insiemi numerabili

Idea: Listiamo per diagonali: I, II, III, IV, V, VI,...

$$\begin{array}{ccccc}
1/1 & 1/2 & 1/3 & 1/4 & \dots \\
2/1 & 2/2 & 2/3 & 2/4 & \dots \\
3/1 & 3/2 & 3/3 & 3/4 & \dots \\
4/1 & 4/2 & 4/3 & 4/4 & \dots \\
\dots & \dots & \dots & \dots & \dots
\end{array}$$

$1/1, 2/1, 1/2, 3/1, 2/2, 1/3, \dots$

Nota: dovremmo eliminare i duplicati (ma è solo una questione tecnica)

Insiemi numerabili

Σ^* **è numerabile.**:
listiamo prima la stringa vuota, poi le stringhe (in ordine lessicografico) lunghe 1, poi 2, $\ldots$

Esempio: $\Sigma = \{0, 1\}$, $w_0 = \epsilon$, $w_1 = 0$, $w_2 = 1$, $w_3 = 00$, ...

Insiemi numerabili

L'insieme

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

è numerabile: è possibile codificare una MdT
M con una stringa su un alfabeto Σ.

L'insieme dei numeri reali non è numerabile.

Sia per assurdo $\mathbb{R}$ numerabile, allora possiamo costruire la lista

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

L'insieme dei numeri reali non è numerabile.

Sia per assurdo $\mathbb{R}$ numerabile, allora possiamo costruire la lista

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

Per ogni $i \geq 1$, scriviamo $f(i) = f_0(i), f_1(i)f_2(i)f_3(i)\ldots$
Es: se $f(1) = 4,256\ldots$
allora $f_0(1) = 4,\ f_1(1) = 2,\ f_1(1) = 2,\ f_2(1) = 5,\ f_3(1) = 6,\ldots$

L'insieme dei numeri reali non è numerabile.

Sia per assurdo $\mathbb{R}$ numerabile, allora possiamo costruire la lista

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

Per ogni $i \geq 1$, scriviamo $f(i) = f_0(i), f_1(i)f_2(i)f_3(i)\dots$
Es: se $f(1) = 4,256\dots$
allora $f_0(1) = 4,\ f_1(1) = 2,\ f_1(1) = 2,\ f_2(1) = 5,\ f_3(1) = 6, \dots$

Organizziamoli in una matrice:

$$\begin{array}{l|cccccc}
i \backslash f(i) & f_1 & f_2 & f_3 & \dots & \dots & \dots \\
\hline
1 & \color{red}{f_1(1)} & f_2(1) & f_3(1) & \dots & f_i(1) & \dots \\
2 & f_1(2) & \color{red}{f_2(2)} & f_3(2) & \dots & f_i(2) & \dots \\
3 & f_1(3) & f_2(3) & \color{red}{f_3(3)} & \dots & f_i(3) & \dots \\
\vdots & \vdots & \vdots & \vdots & \color{red}{\ddots} & \vdots & \vdots \\
i & f_1(i) & f_2(i) & f_3(i) & \dots & \color{red}{f_i(i)} & \dots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \color{red}{\ddots}
\end{array}$$

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	$f_1(2)$	$f_2(2)$	$f_3(2)$	$\ldots$	$f_i(2)$	$\ldots$
3	$f_1(3)$	$f_2(3)$	$f_3(3)$	$\ldots$	$f_i(3)$	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
i	$f_1(i)$	$f_2(i)$	$f_3(i)$	$\ldots$	$f_i(i)$	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Sia

$$x = 0, x_1 x_2 x_3 \ldots x_i \ldots$$

il numero nell'intervallo $(0, 1)$ avente l'i-ma cifra decimale pari a

$$x_i \neq f_i(i) \text{ per ogni } i \geq 1$$

Chiaramente $x \in \mathbb{R}$.

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

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

Sia

$$x = 0, x_1 x_2 x_3 \dots x_i \dots$$

il numero nell'intervallo $(0, 1)$ avente l'i-ma cifra decimale pari a

$$x_i \neq f_i(i) \text{ per ogni } i \geq 1$$

Chiaramente $x \in \mathbb{R}$. Risulta x nella lista?

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

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

Sia

$$x = 0, x_1 x_2 x_3 \dots x_i \dots$$

il numero nell'intervallo $(0, 1)$ avente l'i-ma cifra decimale pari a

$$x_i \neq f_i(i) \text{ per ogni } i \geq 1$$

Chiaramente $x \in \mathbb{R}$. Risulta x nella lista?
Per un qualsiasi intero j, il suo j-mo digit soddisfa $x_j \neq f_j(j)$
quindi deve risultare $x \neq f(j)$!
Quindi $x \in \mathbb{R}$ non può comparire nella lista e $\mathbb{R}$ non numerabile

Il Metodo della diagonalizzazione

$\{w_1, w_2, \ldots\} = \Sigma^*$, $\{M_1, M_2, \ldots\} = \{\text{MdT su } \Sigma\}$: numerabili

Il Metodo della diagonalizzazione

$\{w_1, w_2, \ldots\} = \Sigma^*$, $\{M_1, M_2, \ldots\} = \{\text{MdT su } \Sigma\}$: numerabili

	w_1	w_2	w_3	$\ldots$	w_i	w_j
M_1	$x_{1,1}$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
M_2	$\cdot$	$x_{2,2}$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$x_{3,3}$	$\cdot$	$\cdot$	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$x_{4,4}$	$\cdot$	$\cdot$
M_i	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$x_{i,i}$	$x_{i,j}$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$

con $x_{i,j} = 1$ se $w_j \in L(M_i)$, $x = 0$ altrimenti.

Il Metodo della diagonalizzazione

$\{w_1, w_2, \ldots\} = \Sigma^*$, $\{M_1, M_2, \ldots\} = \{\text{MdT su } \Sigma\}$: numerabili

	w_1	w_2	w_3	$\ldots$	w_i	w_j
M_1	$x_{1,1}$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
M_2	$\cdot$	$x_{2,2}$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$x_{3,3}$	$\cdot$	$\cdot$	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$x_{4,4}$	$\cdot$	$\cdot$
M_i	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$x_{i,i}$	$x_{i,j}$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$

con $x_{i,j} = 1$ se $w_j \in L(M_i)$, $x = 0$ altrimenti.

Sia $L = \{w_i \in \Sigma^* \mid w_i \notin L(M_i)\}$

L è il "complemento della diagonale":

se l'elemento (M_i, w_i) della diagonale è $x_{i,i} = 1$, allora $w_i \notin L$;

se l'elemento (M_i, w_i) della diagonale è $x_{i,i} = 0$, allora $w_i \in L$

Il Metodo della diagonalizzazione

$$
\begin{array}{c|cccccc}
 & w_1 & w_2 & w_3 & \dots & w_i & w_j \\
\hline
M_1 & {\color{red}x_{1,1}} & \cdot & \cdot & \cdot & \cdot & \cdot \\
M_2 & \cdot & {\color{red}x_{2,2}} & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & {\color{red}x_{3,3}} & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & {\color{red}x_{4,4}} & \cdot & \cdot \\
M_i & \cdot & \cdot & \cdot & \cdot & {\color{red}x_{i,i}} & x_{i,j} \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & {\color{red}\cdot} \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{array}
$$

con $x_{i,j} = 1$ se $w_j \in L(M_i)$, $x_{i,j} = 0$ altrimenti.

Sia $L = \{w_i \in \Sigma^* \mid w_i \notin L(M_i)\}$

Può L comparire nella lista?

Il Metodo della diagonalizzazione

	w_1	w_2	w_3	$\dots$	w_i	w_j
M_1	$x_{1,1}$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
M_2	$\cdot$	$x_{2,2}$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$x_{3,3}$	$\cdot$	$\cdot$	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$x_{4,4}$	$\cdot$	$\cdot$
M_i	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$x_{i,i}$	$x_{i,j}$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$

con $x_{i,j} = 1$ se $w_j \in L(M_i)$, $x_{i,j} = 0$ altrimenti.

Sia $L = \{w_i \in \Sigma^* \mid w_i \notin L(M_i)\}$
Può L comparire nella lista?
Supponiamo $L = L(M_h)$,

- $w_h \in L \Rightarrow x_{h,h} = 0 \Rightarrow w_h \notin L(M_h) = L$ contraddizione
- $w_h \notin L \Rightarrow x_{h,h} = 1 \Rightarrow w_h \in L(M_h) = L$ contraddizione

Il Metodo della diagonalizzazione

"Esistono più linguaggi che macchine di Turing"

Il Metodo della diagonalizzazione

"Esistono più linguaggi che macchine di Turing"

Corollary

Esistono linguaggi che non sono Turing riconoscibili

Macchina di Turing Universale

- Una MdT universale U simula la computazione di una qualsiasi MdT M

Macchina di Turing Universale

- Una MdT universale U simula la computazione di una qualsiasi MdT M
- U riceve in input una **rappresentazione** $\langle M, w \rangle$ di M e di un possibile input w di M
  Macchina di Turing Universale
  Ricorda: Abbiamo visto che è possibile codificare una MdT M e una stringa w con una stringa su un alfabeto Σ.
  Es. $\langle M, w \rangle =$ "codifica di $M''\#$"codifica di w''.

Macchina di Turing Universale

- Una MdT universale U simula la computazione di una qualsiasi MdT M
- U riceve in input una **rappresentazione** $\langle M, w\rangle$ di M e di un possibile input w di M
- È chiamata universale perchè la computazione di una qualsiasi MdT può essere simulata da U

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

A_{TM} è Turing riconoscibile

Teorema

Il linguaggio

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

è Turing riconoscibile.

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.

Definiamo una MdT U che accetta A_{TM}: sull'input $\langle M, w\rangle$ dove M è una MdT e w è una stringa

$\square$

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.

Definiamo una MdT U che accetta A_{TM}: sull'input $\langle M, w\rangle$ dove M è una MdT e w è una stringa

1. Simula M sull'input w.

$\square$

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.

Definiamo una MdT U che accetta A_{TM}: sull'input $\langle M, w\rangle$ dove M è una MdT e w è una stringa

1. Simula M sull'input w.
2. Se M accetta, accetta; se M rifiuta, rifiuta.

$\square$

Dettagli: Simulazione di MdT input

Abbiamo visto MdT che simula Automa
Simulare MdT M con altra MdT risulta molto simile.

Dettagli: Simulazione di MdT input

Abbiamo visto MdT che simula Automa
Simulare MdT M con altra MdT risulta molto simile.

1. Marca stato iniziale di M (stato corrente) e primo simbolo su nastro (posizione corrente testina)
2. cerca prossima transizione (nella parte che descrive la funzione di transizione),
   sia $(q, x) \to (q', x', D)$
3. Esegui la transizione
4. Aggiorna lo stato corrente (marca q') e la posizione corrente della testina (marca simbolo a D)
5. Se lo stato corrente risulta q_{accept}/q_{reject} decidi di conseguenza, altrimenti ripeti da 2

Note

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

Note

1. **Nota:** U è detta MdT universale.
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}.

Indecidibilità del problema della fermata

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT e } w \in L(M)\}$$

Teorema

Il linguaggio A_{TM} non è decidibile.

Paradosso di Bertrand Russel

In un paese vive un solo barbiere, un uomo ben sbarbato, che rade tutti e soli gli uomini del villaggio che non si radono da soli.

Chi sbarba il barbiere?

Paradosso di Bertrand Russel

In un paese vive un solo barbiere, un uomo ben sbarbato, che rade tutti e soli gli uomini del villaggio che non si radono da soli.

Chi sbarba il barbiere?

- se il barbiere rade se stesso,

Paradosso di Bertrand Russel

In un paese vive un solo barbiere, un uomo ben sbarbato, che rade tutti e soli gli uomini del villaggio che non si radono da soli.

Chi sbarba il barbiere?

- se il barbiere rade se stesso,
  allora per definizione il barbiere non rade se stesso;

Paradosso di Bertrand Russel

In un paese vive un solo barbiere, un uomo ben sbarbato, che rade tutti e soli gli uomini del villaggio che non si radono da soli.

Chi sbarba il barbiere?

- se il barbiere rade se stesso,
  allora per definizione il barbiere non rade se stesso;
- se il barbiere non rade se stesso

Paradosso di Bertrand Russel

In un paese vive un solo barbiere, un uomo ben sbarbato, che rade tutti e soli gli uomini del villaggio che non si radono da soli.

Chi sbarba il barbiere?

- se il barbiere rade se stesso,
  allora per definizione il barbiere non rade se stesso;
- se il barbiere non rade se stesso
  allora, dato che il barbiere rade tutti quelli che non si radono da soli, il barbiere rade se stesso.

Paradosso di Bertrand Russel

In un paese vive un solo barbiere, un uomo ben sbarbato, che rade tutti e soli gli uomini del villaggio che non si radono da soli.

Chi sbarba il barbiere?

- se il barbiere rade se stesso,
  allora per definizione il barbiere non rade se stesso;
- se il barbiere non rade se stesso
  allora, dato che il barbiere rade tutti quelli che non si radono da soli, il barbiere rade se stesso.

Si tratta di un'antinomia: compresenza di due affermazioni contraddittorie che possono essere entrambe dimostrate o giustificate.
In generale Russel pose il problema dell'insieme di tutti gli insiemi che non contengono se stessi.
Autoreferenza può causare problemi!

Indecidibilità del problema della fermata: Dimostrazione

Supponiamo per assurdo che esiste una macchina di Turing H con due possibili risultati di una computazione (accettazione, rifiuto) e tale che:

$$H = \begin{cases} \textit{accetta} & \text{se } M \text{ accetta } w \\ \textit{rifiuta} & \text{se } M \text{ non accetta } w \end{cases}$$

Indecidibilità del problema della fermata: Dimostrazione

Supponiamo per assurdo che esiste una macchina di Turing H con due possibili risultati di una computazione (accettazione, rifiuto) e tale che:

$$H = \begin{cases} \textit{accetta} & \text{se } M \text{ accetta } w \\ \textit{rifiuta} & \text{se } M \text{ non accetta } w \end{cases}$$

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

Indecidibilità del problema della fermata: Dimostrazione

Costruiamo una nuova MdT D che usa H come sottoprogramma
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*

Indecidibilità del problema della fermata: Dimostrazione

Costruiamo una nuova MdT D che usa H come sottoprogramma
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*

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

Indecidibilità del problema della fermata: Dimostrazione

Costruiamo una nuova MdT D che usa H come sottoprogramma
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*

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

Quindi

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

Indecidibilità del problema della fermata: Dimostrazione

1. **Nota:** MdT M deve essere in grado di accettare ogni stringa.
2. **Nota:** La codifica $\langle M \rangle$ di M è una stringa.

Indecidibilità del problema della fermata: Dimostrazione

1. **Nota:** MdT M deve essere in grado di accettare ogni stringa.
2. **Nota:** La codifica $\langle M \rangle$ di M è una stringa.
3. **Nota:** Operare una macchina sulla sua codifica è analogo ad usare un compilatore Pascal per compilarlo
   (il compilatore Pascal è scritto in Pascal).

Indecidibilità del problema della fermata: Dimostrazione

Autoreferenzialità può essere pericolosa!

Indecidibilità del problema della fermata: Dimostrazione

Autoreferenzialità può essere pericolosa!

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

Indecidibilità del problema della fermata: Dimostrazione

Autoreferenzialità può essere pericolosa!

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

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

Indecidibilità del problema della fermata: Dimostrazione

Autoreferenzialità può essere pericolosa!

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

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

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

Indecidibilità del problema della fermata: Dimostrazione

Autoreferenzialità può essere pericolosa!

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

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

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

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!

$\square$

Indecidibilità del problema della fermata: Riepilogo della Dimostrazione

1. Definiamo $A_{TM} = \{\langle M, w\rangle \mid M$ 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?

Consideriamo la tavola

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

Diagonalizzazione?

Consideriamo H:
MdT H rifiuta anche se M_i va in loop (oltre a se M_i rifiuta)

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

Diagonalizzazione?

Consideriamo ora D e $D(\langle D \rangle)$:

Dobbiamo considerare la diagonale!

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\ldots$	$\langle D \rangle$	$\ldots$
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_2	acc	acc	rej	rej	$\ldots$	$\ldots$	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
D	acc	acc	rej	rej	$\ldots$	???	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Indecidibilità

- Nella prova precedente è stato utilizzato il metodo della diagonalizzazione.

Indecidibilità

- Nella prova precedente è stato utilizzato il metodo della diagonalizzazione.
- In conclusione, A_{TM} è Turing riconoscibile ma è indecidibile.

Indecidibilità

- Nella prova precedente è stato utilizzato il metodo della diagonalizzazione.
- In conclusione, A_{TM} è Turing riconoscibile ma è indecidibile.
- Che differenza c'è tra le due dimostrazioni?
  Cioè che differenza c'è tra U e D?
- Sappiamo che esistono linguaggi che non sono Turing riconoscibili.
- Vogliamo individuare uno specifico linguaggio non Turing riconoscibile ($\overline{A_{TM}}$)

Un linguaggio che non è Turing riconoscibile

Definizione

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

Un linguaggio che non è Turing riconoscibile

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.*

Un linguaggio che non è Turing riconoscibile

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

Un linguaggio che non è Turing riconoscibile

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 $w \notin L$:

Un linguaggio che non è Turing riconoscibile

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 $w \notin L$:

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

Un linguaggio che non è Turing riconoscibile

($\Leftarrow$) 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

Un linguaggio che non è Turing riconoscibile

($\Leftarrow$) 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

Un linguaggio che non è Turing riconoscibile

($\Leftarrow$) 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

Un linguaggio che non è Turing riconoscibile

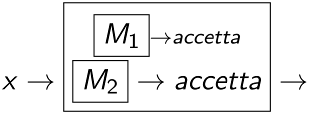
($\Leftarrow$) Supponiamo che L e il suo complemento siano entrambi Turing riconoscibili. Sia M_1 una MdT che riconosce L e M_2 una MdT che riconosce $\overline{L}$. Definiamo una MdT N (a due nastri): sull'input x

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
3. Se M_1 accetta, *accetta*; se M_2 accetta, *rifiuta*

Un linguaggio che non è Turing riconoscibile

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

Un linguaggio che non è Turing riconoscibile



$$\begin{cases} \textit{accetta} & \text{se } M_1 \text{ accetta} \\ \textit{rifiuta} & \text{se } M_2 \text{ accetta} \end{cases}$$

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

Un linguaggio che non è Turing riconoscibile

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

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.

Un linguaggio che non è Turing riconoscibile

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

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.

Un linguaggio che non è Turing riconoscibile

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

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
$x \in L$.

$\square$

Un linguaggio che non è Turing riconoscibile

Teorema

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

Dimostrazione.

$\square$

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.

$\square$

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.

$\square$

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.

$\square$

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.

$\square$

Riducibilità

È importante riconoscere che un problema P è indecidibile.

Riducibilità

È importante riconoscere che un problema P è indecidibile.

Come? Due possibilità:

Riducibilità

È importante riconoscere che un problema P è indecidibile.

Come? Due possibilità:

- Supporre l'esistenza di una MdT
  che decide P e provare che questo conduce a una contraddizione.

Riducibilità

È importante riconoscere che un problema P è indecidibile.

Come? Due possibilità:

- Supporre l'esistenza di una MdT
  che decide P e provare che questo conduce a una contraddizione.
- Considerare un problema P' di cui sia nota l'indecidibilità e dimostrare che P' "non è più difficile" del problema in questione P.

Riducibilità: un esempio

Esempio $\Sigma = \{0, 1\}$.

Riducibilità: un esempio

Esempio $\Sigma = \{0, 1\}$.

$EVEN = \{w \in \Sigma^* \mid w \text{ è la rappresentazione binaria di } n \in \mathbb{N} \text{ pari}\}$

Riducibilità: un esempio

Esempio $\Sigma = \{0, 1\}$.

$$EVEN = \{w \in \Sigma^* \mid w \text{ è la rappresentazione binaria di } n \in \mathbb{N} \text{ pari}\}$$

$$ODD = \{w \in \Sigma^* \mid w \text{ è la rappresentazione binaria di } n \in \mathbb{N} \text{ dispari}\}$$

Riducibilità: un esempio

Esempio $\Sigma = \{0, 1\}$.

$$EVEN = \{w \in \Sigma^* \mid w \text{ è la rappresentazione binaria di } n \in \mathbb{N} \text{ pari}\}$$

$$ODD = \{w \in \Sigma^* \mid w \text{ è la rappresentazione binaria di } n \in \mathbb{N} \text{ dispari}\}$$

Sia $w \in \Sigma^*$ e sia n il corrispondente decimale di w. È facile costruire la MdT
INCR:

$$w \rightarrow \boxed{INCR} \rightarrow w' \quad (= \text{ rappresentazione binaria di } n+1)$$

Riducibilità: un esempio

- *EVEN* "non è più difficile" di *ODD*: se esiste una MdT
  R che decide *ODD*, la MdT
  S decide *EVEN*.

Riducibilità: un esempio

- *EVEN* "non è più difficile" di *ODD*: se esiste una MdT R che decide *ODD*, la MdT S decide *EVEN*.

$$S : w \rightarrow \boxed{INCR} \rightarrow w' \rightarrow \boxed{R}$$

Riducibilità: un esempio

- *EVEN* "non è più difficile" di *ODD*: se esiste una MdT
  R che decide *ODD*, la MdT
  S decide *EVEN*.

$$S : w \rightarrow \boxed{INCR} \rightarrow w' \rightarrow \boxed{R}$$

- Viceversa se *EVEN* è indecidibile proviamo così che anche
  ODD lo è: se per assurdo esistesse una MdT
  R che decide *ODD*, la MdT
  S deciderebbe *EVEN*.

Riducibilità: un esempio

- *EVEN* "non è più difficile" di *ODD*: se esiste una MdT R che decide *ODD*, la MdT S decide *EVEN*.

$$S : w \to \boxed{INCR} \to w' \to \boxed{R}$$

- Viceversa se *EVEN* è indecidibile proviamo così che anche *ODD* lo è: se per assurdo esistesse una MdT R che decide *ODD*, la MdT S deciderebbe *EVEN*.
- Problema. Possiamo dire che *ODD* "non è più difficile" di *EVEN*? In che modo?

Riducibilità: un altro esempio

Esempio (il “vero” problema della fermata):

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT e } M \text{ si arresta su } w\}$$

Riducibilità: un altro esempio

Esempio (il "vero" problema della fermata):

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT e } M \text{ si arresta su } w\}$$

Se $HALT_{TM}$ fosse decidibile potremmo decidere anche A_{TM}:

Riducibilità: un altro esempio

Esempio (il "vero" problema della fermata):

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT e } M \text{ si arresta su } w\}$$

Se $HALT_{TM}$ fosse decidibile potremmo decidere anche A_{TM}:

- Sia R una MdT che decide $HALT_{TM}$.

Riducibilità: un altro esempio

Esempio (il "vero" problema della fermata):

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT e } M \text{ si arresta su } w\}$$

Se $HALT_{TM}$ fosse decidibile potremmo decidere anche A_{TM}:

- Sia R una MdT che decide $HALT_{TM}$.
- Costruiamo S (che decide A_{TM}) che sull'input $\langle M, w\rangle$, dove M è una MdT e w è una stringa:

Riducibilità: un altro esempio

Esempio (il "vero" problema della fermata):

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT e } M \text{ si arresta su } w\}$$

Se $HALT_{TM}$ fosse decidibile potremmo decidere anche A_{TM}:

- Sia R una MdT che decide $HALT_{TM}$.
- Costruiamo S (che decide A_{TM}) che sull'input $\langle M, w\rangle$, dove M è una MdT e w è una stringa:
  - simula R su $\langle M, w\rangle$

Riducibilità: un altro esempio

Esempio (il "vero" problema della fermata):

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT e } M \text{ si arresta su } w\}$$

Se $HALT_{TM}$ fosse decidibile potremmo decidere anche A_{TM}:

- Sia R una MdT che decide $HALT_{TM}$.
- Costruiamo S (che decide A_{TM}) che sull'input $\langle M, w\rangle$, dove M è una MdT e w è una stringa:
  - simula R su $\langle M, w\rangle$
  - se R rifiuta, allora S rifiuta (poichè M va in loop $w \notin L(M)$);
    se R accetta (questo significa che M si ferma su w) allora simula M finchè M si arresta su w.

Riducibilità: un altro esempio

Esempio (il "vero" problema della fermata):

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT e } M \text{ si arresta su } w\}$$

Se $HALT_{TM}$ fosse decidibile potremmo decidere anche A_{TM}:

- Sia R una MdT che decide $HALT_{TM}$.
- Costruiamo S (che decide A_{TM}) che sull'input $\langle M, w\rangle$, dove M è una MdT e w è una stringa:
  - simula R su $\langle M, w\rangle$
  - se R rifiuta, allora S rifiuta (poichè M va in loop $w \notin L(M)$); se R accetta (questo significa che M si ferma su w) allora simula M finchè M si arresta su w.
  - Se M ha accettato, accetta ($w \in L(M)$); se M ha rifiutato, rifiuta ($w \notin L(M)$).

Riducibilità: un altro esempio

Esempio (il "vero" problema della fermata):

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT e } M \text{ si arresta su } w\}$$

Se $HALT_{TM}$ fosse decidibile potremmo decidere anche A_{TM}:

- Sia R una MdT che decide $HALT_{TM}$.
- Costruiamo S (che decide A_{TM}) che sull'input $\langle M, w\rangle$, dove M è una MdT e w è una stringa:
  - simula R su $\langle M, w\rangle$
  - se R rifiuta, allora S rifiuta (poichè M va in loop $w \notin L(M)$); se R accetta (questo significa che M si ferma su w) allora simula M finchè M si arresta su w.
  - Se M ha accettato, accetta ($w \in L(M)$); se M ha rifiutato, rifiuta ($w \notin L(M)$).

Se esistesse R che decide $HALT_{TM}$ allora otterremmo S che decide A_{TM}. Poichè sappiamo che A_{TM} è indecidibile allora R non può esistere e $HALT_{TM}$ deve essere indecidibile.

Schema di Riduzione

Dal problema A **al** Problema B

1. Sappiamo che A risulta indecidibile

Schema di Riduzione

Dal problema A **al** Problema B

1. Sappiamo che A risulta indecidibile
2. Vogliamo provare che B è indecidibile

Schema di Riduzione

Dal problema A **al** Problema B

1. Sappiamo che A risulta indecidibile
2. Vogliamo provare che B è indecidibile
3. Assumiamo (per assurdo) B decidibile ed
   usiamo questa assunzione per provare A decidibile

Schema di Riduzione

Dal problema A **al** Problema B

1. Sappiamo che A risulta indecidibile
2. Vogliamo provare che B è indecidibile
3. Assumiamo (per assurdo) B decidibile ed usiamo questa assunzione per provare A decidibile
4. La contraddizione ci fa concludere: B indecidibile

Riepilogo per $HALT_{TM}$

Dal problema A **al** Problema B

1. Sappiamo che A risulta indecidibile
   Unico conosciuto: $A = A_{TM}$

Riepilogo per $HALT_{TM}$

Dal problema A **al** Problema B

1. Sappiamo che A risulta indecidibile
   Unico conosciuto: $A = A_{TM}$
2. Vogliamo provare che B è indecidibile
   $HALT_{TM}$ gioca il ruolo di B

Riepilogo per $HALT_{TM}$

Dal problema A **al** Problema B

1. Sappiamo che A risulta indecidibile
   Unico conosciuto: $A = A_{TM}$
2. Vogliamo provare che B è indecidibile
   $HALT_{TM}$ gioca il ruolo di B
3. Assumiamo (per assurdo) B decidibile ed usiamo questa assunzione per provare A decidibile
   Proviamo $HALT_{TM}$ decidibile $\Rightarrow$ A_{TM} decidibile
   Sia R una MdT che decide $HALT_{TM}$.
   Costruiamo S che sull'input $\langle M, w \rangle$
   - simula R su $\langle M, w \rangle$
   - Se R accetta (M si ferma su w)
     - simula M finchè M si arresta su w.
       Se M ha accettato, accetta; se M ha rifiutato, rifiuta.

   Se R decide $HALT_{TM}$ allora S decide A_{TM}; impossibile!!

Riepilogo per $HALT_{TM}$

Dal problema A **al** Problema B

1. Sappiamo che A risulta indecidibile
   Unico conosciuto: $A = A_{TM}$
2. Vogliamo provare che B è indecidibile
   $HALT_{TM}$ gioca il ruolo di B
3. Assumiamo (per assurdo) B decidibile ed usiamo questa assunzione per provare A decidibile
   Proviamo $HALT_{TM}$ decidibile $\Rightarrow$ A_{TM} decidibile
   Sia R una MdT che decide $HALT_{TM}$.
   Costruiamo S che sull'input $\langle M, w\rangle$
   - simula R su $\langle M, w\rangle$
   - Se R accetta (M si ferma su w)
     - simula M finchè M si arresta su w.
       Se M ha accettato, accetta; se M ha rifiutato, rifiuta.

   Se R decide $HALT_{TM}$ allora S decide A_{TM}; impossibile!!
4. La contraddizione ci fa concludere: $B = HALT_{TM}$ indecidibile

Problema del vuoto di MdT

Consideriamo

$$E_{TM} = \{\langle M \rangle \mid M \text{ MdT tale che } L(M) = \emptyset\}$$

Teorema

E_{TM} è indecidibile.

1. Sappiamo che A_{TM} risulta indecidibile

Problema del vuoto di MdT

Consideriamo

$$E_{TM} = \{\langle M \rangle \mid M \text{ MdT tale che } L(M) = \emptyset\}$$

Teorema

E_{TM} è indecidibile.

1. Sappiamo che A_{TM} risulta indecidibile
2. Vogliamo provare che E_{TM} è indecidibile

Problema del vuoto di MdT

Consideriamo

$$E_{TM} = \{\langle M\rangle \mid M \text{ MdT tale che } L(M) = \emptyset\}$$

Teorema

E_{TM} è indecidibile.

1. Sappiamo che A_{TM} risulta indecidibile
2. Vogliamo provare che E_{TM} è indecidibile
3. Assumiamo (per assurdo) E_{TM} decidibile ed
   usiamo questa assunzione per provare A_{TM} decidibile

Problema del vuoto di MdT

Consideriamo

$$E_{TM} = \{\langle M \rangle \mid M \text{ MdT tale che } L(M) = \emptyset\}$$

Teorema

E_{TM} è indecidibile.

1. Sappiamo che A_{TM} risulta indecidibile
2. Vogliamo provare che E_{TM} è indecidibile
3. Assumiamo (per assurdo) E_{TM} decidibile ed usiamo questa assunzione per provare A_{TM} decidibile
4. La contraddizione ci fa concludere: E_{TM} indecidibile

Problema del vuoto di MdT

Assumiamo (per assurdo) E_{TM} decidibile, sia R MdT che lo decide
Usiamo R per costruire una MdT S che decide A_{TM}:

Problema del vuoto di MdT

Assumiamo (per assurdo) E_{TM} decidibile, sia R MdT che lo decide
Usiamo R per costruire una MdT S che decide A_{TM}:

Data istanza $\langle M, w\rangle$ di A_{TM},
Usiamo R su $\langle M\rangle$.

R accetta $\langle M\rangle \Rightarrow L(M) = \emptyset \Rightarrow M$ non accetta w
$\Rightarrow$ Decider S per A_{TM} deve rifiutare $\langle M, w\rangle$.

Problema del vuoto di MdT

Assumiamo (per assurdo) E_{TM} decidibile, sia R MdT che lo decide
Usiamo R per costruire una MdT S che decide A_{TM}:

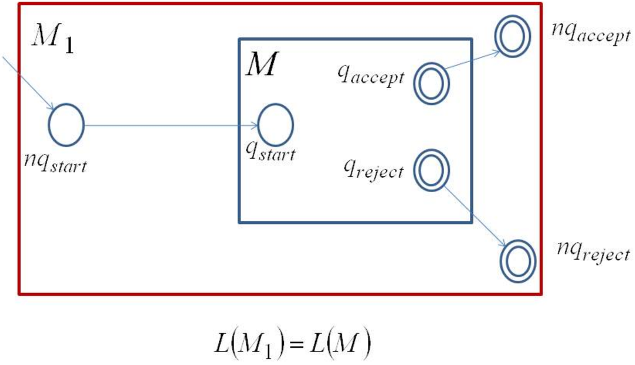
Data istanza $\langle M, w \rangle$ di A_{TM},
Usiamo R su $\langle M \rangle$.

R accetta $\langle M \rangle \Rightarrow L(M) = \emptyset \Rightarrow M$ non accetta w
$\Rightarrow$ Decider S per A_{TM} deve rifiutare $\langle M, w \rangle$.

R rifiuta $\langle M \rangle \Rightarrow L(M) \neq \emptyset$
Rimane la domanda: $w \in L(M)$?
Modifichiamo M in M_1

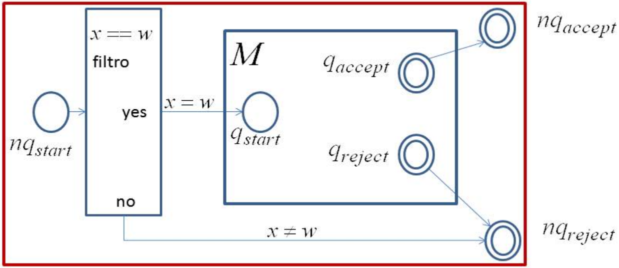
Problema del vuoto di MdT: M_1

Iniziamo con MdT t.c. $L(M_1) = L(M)$
(Nota: le nuove transizioni hanno label $x- > x, S$, per ogni $x \in \Gamma$)



Problema del vuoto di MdT: M_1

Inseriamo filtro che controlla se l'input corrisponde a w:
facendo un confronto carattere per carattere tra input x e stringa w (data)



$$L(M_1) = \begin{cases} \{w\} & \text{se } M \text{ accetta } w \\ \phi & \text{se } M \text{ rifiuta } w \end{cases}$$

Problema del vuoto di MdT: M_1

Descrizione formale di M_1

M_1 su input x

1. Se $x \neq w$, rifiuta
2. Se $x = w$ e M accetta w, accetta

M rifiuta w sse $L(M_1) = \emptyset$

Problema del vuoto di MdT: S

Input di S corrisponde alla coppia $\langle M, w \rangle$
Prima di "usare" R (decider di E_{TM}), S deve calcolare la codifica $\langle M_1 \rangle$ di M_1 (deve aggiungere il filtro)

Problema del vuoto di MdT: S

Input di S corrisponde alla coppia $\langle M, w \rangle$
Prima di "usare" R (decider di E_{TM}), S deve calcolare la codifica $\langle M_1 \rangle$ di M_1 (deve aggiungere il filtro)

S su input $\langle M, w \rangle$

1. Calcola la codifica $\langle M_1 \rangle$ di M_1
2. Usa R su input $\langle M_1 \rangle$
3. Se R rifiuta, accetta
   se R accetta ($L(M_1) = \emptyset$), rifiuta

Problema del vuoto di MdT: S

Input di S corrisponde alla coppia $\langle M, w \rangle$
Prima di "usare" R (decider di E_{TM}), S deve calcolare la codifica $\langle M_1 \rangle$ di M_1 (deve aggiungere il filtro)

S su input $\langle M, w \rangle$

1. Calcola la codifica $\langle M_1 \rangle$ di M_1
2. Usa R su input $\langle M_1 \rangle$
3. Se R rifiuta, accetta
   se R accetta ($L(M_1) = \emptyset$), rifiuta

Se R accetta ($L(M_1) = \emptyset$ e quindi $w \notin L(M)$) quindi S rifiuta

Problema del vuoto di MdT: S

Input di S corrisponde alla coppia $\langle M, w \rangle$
Prima di "usare" R (decider di E_{TM}), S deve calcolare la codifica $\langle M_1 \rangle$ di M_1 (deve aggiungere il filtro)

S su input $\langle M, w \rangle$

1. Calcola la codifica $\langle M_1 \rangle$ di M_1
2. Usa R su input $\langle M_1 \rangle$
3. Se R rifiuta, accetta
   se R accetta ($L(M_1) = \emptyset$), rifiuta

Se R accetta ($L(M_1) = \emptyset$ e quindi $w \notin L(M)$) quindi S rifiuta

Se R rifiuta ($L(M_1) = \{w\}$ e $w \in L(M)$) quindi S accetta

Conclusione: Da R abbiamo costruito un decider S per A_{TM}, quindi R non esiste.

Funzioni calcolabili

Definizione

Una funzione $f : \Sigma^* \to \Sigma^*$ *è calcolabile se esiste una TM* M *tale che su ogni input* w, M *si arresta con* $f(w)$, *e solo con* $f(w)$, *sul suo nastro.*

Funzioni calcolabili

Definizione

Una funzione $f : \Sigma^* \to \Sigma^*$ *è calcolabile se esiste una TM* M *tale che su ogni input* w*,* M *si arresta con* $f(w)$*, e solo con* $f(w)$*, sul suo nastro.*

- Nota: questa definizione sottolinea la differenza tra definire una funzione f, cioè definire i valori di f e calcolare tali valori di f.

Funzioni calcolabili

Definizione

Una funzione $f : \Sigma^* \to \Sigma^*$ è calcolabile se esiste una TM M tale che su ogni input w, M si arresta con $f(w)$, e solo con $f(w)$, sul suo nastro.

- Nota: questa definizione sottolinea la differenza tra definire una funzione f, cioè definire i valori di f e calcolare tali valori di f.
- La funzione $F : \langle M, w \rangle \to \langle M_1, w \rangle$ dell'esempio precedente è calcolabile.

Funzioni calcolabili

Le seguenti funzioni aritmetiche sono calcolabili (dove $n, m \in \mathbb{N}$):

- $incr(n) = n + 1$
- $dec(n) = \begin{cases} n - 1 & \text{se } n > 0; \\ 0 & \text{se } n = 0 \end{cases}$
- $(m, n) \to m + n;$
- $(m, n) \to m - n;$
- $(m, n) \to m \cdot n$

Riducibilità

Definizione

Un linguaggio A è riducibile a un linguaggio B ($A \leq_m B$) se esiste una funzione calcolabile $f : \Sigma^* \rightarrow \Sigma^*$ tale che $\forall w$

$$w \in A \Leftrightarrow f(w) \in B$$

La funzione f è chiamata la riduzione di A a B.

Riducibilità

Definizione

Un linguaggio A è riducibile a un linguaggio B ($A \leq_m B$) se esiste una funzione calcolabile $f : \Sigma^* \rightarrow \Sigma^*$ tale che $\forall w$

$$w \in A \Leftrightarrow f(w) \in B$$

La funzione f è chiamata la riduzione di A a B.

- Una riduzione fornisce un modo per convertire problemi di appartenenza ad A in problemi di appartenenza a B.
- Se un problema A è riducibile a B e sappiamo risolvere B allora sappiamo risolvere A
  $\Rightarrow$ A "non è più difficile" di B.

Riduzioni

Teorema

Se $A \leq_m B$ e B è decidibile, allora A è decidibile.

Riduzioni

Teorema

Se $A \leq_m B$ e B è decidibile, allora A è decidibile.

Siano: M il decider per B ed f la riduzione da A a B
Costruiamo un decider N per A: su input w

- Calcola $f(w)$
- "utilizza" M su $f(w)$ e da lo stesso output.

$w \in A \Leftrightarrow f(w) \in B$ (f riduzione da A a B) $\Leftrightarrow M$ accetta $f(w)$

Quindi N decide A.

Riduzioni

Teorema

Se $A \leq_m B$ e B è decidibile, allora A è decidibile.

Siano: M il decider per B ed f la riduzione da A a B
Costruiamo un decider N per A: su input w

- Calcola $f(w)$
- "utilizza" M su $f(w)$ e da lo stesso output.

$w \in A \Leftrightarrow f(w) \in B$ (f riduzione da A a B) $\Leftrightarrow M$ accetta $f(w)$

Quindi N decide A.

Teorema

Se $A \leq_m B$ e B è Turing riconoscibile, allora A è Turing riconoscibile.

R_A riconoscitore per A, R_B riconoscitore per B,

$$R_A : w \rightarrow \boxed{f} \rightarrow f(w) \rightarrow \boxed{R_B}$$

Riduzioni

Corollario

Se $A \leq_m B$ e A è indecidibile, allora B è indecidibile.

Riduzioni

Corollario
Se $A \leq_m B$ e A è indecidibile, allora B è indecidibile.

(se B fosse decidibile lo sarebbe anche A in virtù del teorema precedente)

Riduzioni

Corollario

Se $A \leq_m B$ e A è indecidibile, allora B è indecidibile.

(se B fosse decidibile lo sarebbe anche A in virtù del teorema precedente)

Corollario

Se $A \leq_m B$ e A non è Turing riconoscibile, allora B non è Turing riconoscibile.

Riduzioni

Corollario
Se $A \leq_m B$ e A è indecidibile, allora B è indecidibile.

(se B fosse decidibile lo sarebbe anche A in virtù del teorema precedente)

Corollario
Se $A \leq_m B$ e A non è Turing riconoscibile, allora B non è Turing riconoscibile.

(se B fosse Turing riconoscibile lo sarebbe anche A in virtù del teorema precedente)

Esempi di riduzioni

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ è una TM e } w \in L(M)\}$$

$$E_{TM} = \{\langle M\rangle \mid M \text{ è una } TM \text{ e } L(M) = \emptyset\}$$

$$A_{TM} \leq_m \overline{E_{TM}}$$

Esempi di riduzioni

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ è una TM e } w \in L(M)\}$$

$$E_{TM} = \{\langle M\rangle \mid M \text{ è una } TM \text{ e } L(M) = \emptyset\}$$

$$A_{TM} \leq_m \overline{E_{TM}}$$

Consideriamo $f : \Sigma^* \to \Sigma^*$ tale che $f(\langle M, w\rangle) = \langle M_1\rangle$ dove M_1 su un input x:

1. Se $x \neq w$ allora M_1 si ferma e rifiuta x.
2. Se $x = w$ allora M_1 simula M su w e accetta x se M accetta w.

f è una riduzione di A_{TM} a $\overline{E_{TM}}$.

Esempi di riduzioni

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ è una TM e } w \in L(M)\}$$

$$E_{TM} = \{\langle M\rangle \mid M \text{ è una } TM \text{ e } L(M) = \emptyset\}$$

$$A_{TM} \leq_m \overline{E_{TM}}$$

Consideriamo $f : \Sigma^* \to \Sigma^*$ tale che $f(\langle M, w\rangle) = \langle M_1\rangle$ dove M_1 su un input x:

1. Se $x \neq w$ allora M_1 si ferma e rifiuta x.
2. Se $x = w$ allora M_1 simula M su w e accetta x se M accetta w.

f è una riduzione di A_{TM} a $\overline{E_{TM}}$.
Dimostrazione: – La funzione f è calcolabile
– M accetta w (cioè $\langle M, w\rangle \in A_{TM}$) se e solo se $L(M_1) \neq \emptyset$ (cioè se e solo se $\langle M_1\rangle \in \overline{E_{TM}}$).

Esempi di riduzioni

$A_{TM} \leq_m \overline{E_{TM}}$ e A_{TM} indecidibile $\Rightarrow$ $\overline{E_{TM}}$ indecidibile.

Quindi anche E_{TM} è indecidibile
(decidibilità non risente della complementazione)

Nota. Non si conosce una riduzione da A_{TM} a E_{TM}.

Esempi di riduzioni

$$REGULAR_{TM} = \{\langle M \rangle \mid M \text{ è una } MdT \text{ e } L(M) \text{ è regolare } \}$$

$$A_{TM} \leq_m REGULAR_{TM}$$

Esempi di riduzioni

$$REGULAR_{TM} = \{\langle M \rangle \mid M \text{ è una } MdT \text{ e } L(M) \text{ è regolare } \}$$

$$A_{TM} \leq_m REGULAR_{TM}$$

$f : \langle M, w \rangle \rightarrow \langle R \rangle$ è riduzione da A_{TM} a $REGULAR_{TM}$
dove R su un input x:

1. Se $x \in \{0^n 1^n \mid n \in \mathbb{N}\}$, allora R si ferma e accetta x.
2. Se $x \notin \{0^n 1^n \mid n \in \mathbb{N}\}$ allora R simula M su w e accetta x se M accetta w.

Esempi di riduzioni

$$REGULAR_{TM} = \{\langle M \rangle \mid M \text{ è una } MdT \text{ e } L(M) \text{ è regolare } \}$$

$$A_{TM} \leq_m REGULAR_{TM}$$

$f : \langle M, w \rangle \to \langle R \rangle$ è riduzione da A_{TM} a $REGULAR_{TM}$
dove R su un input x:

1. Se $x \in \{0^n 1^n \mid n \in \mathbb{N}\}$, allora R si ferma e accetta x.
2. Se $x \notin \{0^n 1^n \mid n \in \mathbb{N}\}$ allora R simula M su w e accetta x se M accetta w.

Dimostrazione:
– f è calcolabile (perchè?)
– $L(R) = \Sigma^*$ (regolare)b se M accetta w ed
$L(R) = \{0^n 1^n \mid n \in \mathbb{N}\}$ (non regolare) altrimenti.

Esempi di riduzioni

$$REGULAR_{TM} = \{\langle M \rangle \mid M \text{ è una } MdT \text{ e } L(M) \text{ è regolare } \}$$

$$A_{TM} \leq_m REGULAR_{TM}$$

$f : \langle M, w \rangle \to \langle R \rangle$ è riduzione da A_{TM} a $REGULAR_{TM}$
dove R su un input x:

1. Se $x \in \{0^n 1^n \mid n \in \mathbb{N}\}$, allora R si ferma e accetta x.
2. Se $x \notin \{0^n 1^n \mid n \in \mathbb{N}\}$ allora R simula M su w e accetta x se M accetta w.

Dimostrazione:
– f è calcolabile (perchè?)
– $L(R) = \Sigma^*$ (regolare)b se M accetta w ed
$L(R) = \{0^n 1^n \mid n \in \mathbb{N}\}$ (non regolare) altrimenti.

Quindi M accetta w (cioè $\langle M, w \rangle \in A_{TM}$) se e solo se $L(R)$ è regolare (cioè se e solo se $\langle R \rangle \in REGULAR_{TM}$).

Esempi di riduzioni

$$E_{TM} = \{\langle M \rangle \mid M \text{ è una } MdT \text{ e } L(M) = \emptyset\}$$

$$EQ_{TM} = \{\langle M_1, M_2 \rangle \mid M_1, M_2 \text{ sono MdT e } L(M_1) = L(M_2)\}$$

$$E_{TM} \leq_m EQ_{TM}$$

Esempi di riduzioni

$$E_{TM} = \{\langle M \rangle \mid M \text{ è una } MdT \text{ e } L(M) = \emptyset\}$$

$$EQ_{TM} = \{\langle M_1, M_2 \rangle \mid M_1, M_2 \text{ sono MdT e } L(M_1) = L(M_2)\}$$

$$E_{TM} \leq_m EQ_{TM}$$

Sia M_1 una macchina di Turing tale che $L(M_1) = \emptyset$.
$f : \langle M \rangle \rightarrow \langle M, M_1 \rangle$ è una riduzione di E_{TM} a EQ_{TM}.

Esempi di riduzioni

$$E_{TM} = \{\langle M \rangle \mid M \text{ è una } MdT \text{ e } L(M) = \emptyset\}$$

$$EQ_{TM} = \{\langle M_1, M_2 \rangle \mid M_1, M_2 \text{ sono MdT e } L(M_1) = L(M_2)\}$$

$$E_{TM} \leq_m EQ_{TM}$$

Sia M_1 una macchina di Turing tale che $L(M_1) = \emptyset$.
$f : \langle M \rangle \to \langle M, M_1 \rangle$ è una riduzione di E_{TM} a EQ_{TM}.
Perchè?

Esempi di riduzioni

$$A_{TM} \leq_m EQ_{TM}$$

Esempi di riduzioni

$$A_{TM} \leq_m EQ_{TM}$$

Idea: Data $\langle M, w \rangle$, considerare le MdT $_1$ e M_2 tali che

Esempi di riduzioni

$$A_{TM} \leq_m EQ_{TM}$$

Idea: Data $\langle M, w \rangle$, considerare le MdT $_1$ e M_2 tali che

Per ogni input x:

M_1 accetta x,

M_2 simula M su w. Se M accetta, M_2 accetta.

Esempi di riduzioni

$$A_{TM} \leq_m EQ_{TM}$$

Idea: Data $\langle M, w \rangle$, considerare le MdT $_1$ e M_2 tali che

Per ogni input x:
M_1 accetta x,
M_2 simula M su w. Se M accetta, M_2 accetta.

- $f : \langle M, w \rangle \to \langle M_1, M_2 \rangle$ è riduzione da A_{TM} a EQ_{TM}.
  Perchè?

Esempi di riduzioni

- $A_{TM} \leq_m \overline{EQ_{TM}}$

Esempi di riduzioni

- $A_{TM} \leq_m \overline{EQ_{TM}}$
- Idea: Data $\langle M, w \rangle$, considerare una MT M_1 che **accetta l'insieme vuoto** e una macchina M_2 che accetta Σ^* se M accetta w:

Esempi di riduzioni

- $A_{TM} \leq_m \overline{EQ_{TM}}$
- Idea: Data $\langle M, w \rangle$, considerare una MT M_1 che **accetta l'insieme vuoto** e una macchina M_2 che accetta Σ^* se M accetta w:

  Per ogni input x:
  M_1 rifiuta x,
  M_2 simula M su w. Se M accetta, M_2 accetta.

Esempi di riduzioni

- $A_{TM} \leq_m \overline{EQ_{TM}}$
- Idea: Data $\langle M, w \rangle$, considerare una MT M_1 che **accetta l'insieme vuoto** e una macchina M_2 che accetta Σ^* se M accetta w:

  Per ogni input x:
  M_1 rifiuta x,
  M_2 simula M su w. Se M accetta, M_2 accetta.
- $f : \langle M, w \rangle \rightarrow \langle M_1, M_2 \rangle$ è riduzione da A_{TM} a $\overline{EQ_{TM}}$.

Esempi di riduzioni

- $A_{TM} \leq_m \overline{EQ_{TM}}$
- Idea: Data $\langle M, w \rangle$, considerare una MT M_1 che **accetta l'insieme vuoto** e una macchina M_2 che accetta Σ^* se M accetta w:

  Per ogni input x:
  M_1 rifiuta x,
  M_2 simula M su w. Se M accetta, M_2 accetta.
- $f : \langle M, w \rangle \rightarrow \langle M_1, M_2 \rangle$ è riduzione da A_{TM} a $\overline{EQ_{TM}}$.
- Perchè?

Esempi di riduzioni

Teorema

EQ_{TM} non è nè Turing riconoscibile nè co-Turing riconoscibile.

Dimostrazione.

Supponiamo per assurdo che EQ_{TM} sia Turing riconoscibile.

$A_{TM} \leq_m \overline{EQ_{TM}} \Rightarrow \overline{A_{TM}} \leq_m EQ_{TM}$

Quindi $\overline{A_{TM}}$ sarebbe Turing riconoscibile: assurdo.

Supponiamo per assurdo che EQ_{TM} sia co-Turing riconoscibile, cioè che $\overline{EQ_{TM}}$ sia Turing riconoscibile.

$A_{TM} \leq_m EQ_{TM} \Rightarrow \overline{A_{TM}} \leq_m \overline{EQ_{TM}}$

Quindi $\overline{A_{TM}}$ sarebbe Turing riconoscibile: assurdo.

$\square$

Teorema di Rice

Teorema di Rice. Sia

$$L_{\mathcal{P}} = \{\langle M \rangle \mid M \text{ è una MdT che verifica la proprietà } \mathcal{P}\}$$

un linguaggio che soddisfa le seguenti due condizioni:

Teorema di Rice

Teorema di Rice. Sia

$$L_{\mathcal{P}} = \{\langle M \rangle \mid M \text{ è una MdT che verifica la proprietà } \mathcal{P}\}$$

un linguaggio che soddisfa le seguenti due condizioni:

1. L'appartenenza di M a P dipende solo da $L(M)$, cioè

   $$\forall M_1, M_2 \text{ MdT tali che } L(M_1) = L(M_2), \langle M_1 \rangle \in L_{\mathcal{P}} \Leftrightarrow \langle M_2 \rangle \in L_{\mathcal{P}}$$

Teorema di Rice

Teorema di Rice. Sia

$$L_{\mathcal{P}} = \{\langle M \rangle \mid M \text{ è una MdT che verifica la proprietà } \mathcal{P}\}$$

un linguaggio che soddisfa le seguenti due condizioni:

1. L'appartenenza di M a P dipende solo da $L(M)$, cioè

   $$\forall M_1, M_2 \text{ MdT tali che } L(M_1) = L(M_2), \langle M_1 \rangle \in L_{\mathcal{P}} \Leftrightarrow \langle M_2 \rangle \in L_{\mathcal{P}}$$

2. $\mathcal{P}$ è una proprietà non banale, cioè

   $$\exists M_1, M_2 \text{ MdT tali che } \langle M_1 \rangle \in L_{\mathcal{P}}, \langle M_2 \rangle \notin L_{\mathcal{P}},$$

Teorema di Rice

Teorema di Rice. Sia

$$L_{\mathcal{P}} = \{\langle M \rangle \mid M \text{ è una MdT che verifica la proprietà } \mathcal{P}\}$$

un linguaggio che soddisfa le seguenti due condizioni:

1. L'appartenenza di M a P dipende solo da $L(M)$, cioè

   $$\forall M_1, M_2 \text{ MdT tali che } L(M_1) = L(M_2), \langle M_1 \rangle \in L_{\mathcal{P}} \Leftrightarrow \langle M_2 \rangle \in L_{\mathcal{P}}$$

2. $\mathcal{P}$ è una proprietà non banale, cioè

   $$\exists M_1, M_2 \text{ MdT tali che } \langle M_1 \rangle \in L_{\mathcal{P}}, \langle M_2 \rangle \notin L_{\mathcal{P}},$$

$L_{\mathcal{P}}$ è indecidibile.

Teorema di Rice

- Ogni proprietà non banale del linguaggio di una MdT è indecidibile.

Teorema di Rice

- Ogni proprietà non banale del linguaggio di una MdT è indecidibile.
- Nota la differenza tra una proprietà di $L(M)$ e una proprietà di M:

Teorema di Rice

- Ogni proprietà non banale del linguaggio di una MdT è indecidibile.
- Nota la differenza tra una proprietà di $L(M)$ e una proprietà di M:
  - **Esempio:** $L(M) = \emptyset$ è una proprietà del linguaggio.

Teorema di Rice

- Ogni proprietà non banale del linguaggio di una MdT è indecidibile.
- Nota la differenza tra una proprietà di $L(M)$ e una proprietà di M:
  - **Esempio:** $L(M) = \emptyset$ è una proprietà del linguaggio.
  - **Esempio:** "M ha almeno 1000 stati" è una proprietà della MdT.

Teorema di Rice

- Ogni proprietà non banale del linguaggio di una MdT è indecidibile.
- Nota la differenza tra una proprietà di $L(M)$ e una proprietà di M:
  - **Esempio:** $L(M) = \emptyset$ è una proprietà del linguaggio.
  - **Esempio:** "M ha almeno 1000 stati" è una proprietà della MdT.
  - "$L(M) = \emptyset$" è indecidibile; "M ha almeno 1000 stati" è facilmente decidibile, basta guardare alla codifica di M e contare.

Conseguenze del Teorema di Rice

Non possiamo decidere se una MdT:

- Accetta $\emptyset$.

Conseguenze del Teorema di Rice

Non possiamo decidere se una MdT:

- Accetta $\emptyset$.
- Accetta un linguaggio finito.

Conseguenze del Teorema di Rice

Non possiamo decidere se una MdT:

- Accetta $\emptyset$.
- Accetta un linguaggio finito.
- Accetta un linguaggio regolare, ecc.

Dimostrazione del Teorema di Rice

Mostriamo $A_{TM} \leq_m L_{\mathcal{P}}$

- Sia $T_\emptyset$ una TM tale che $L(T_\emptyset) = \emptyset$;

Dimostrazione del Teorema di Rice

Mostriamo $A_{TM} \leq_m L_{\mathcal{P}}$

- Sia $T_\emptyset$ una TM tale che $L(T_\emptyset) = \emptyset$;
- Possiamo assumere $\langle T_\emptyset \rangle \notin L_{\mathcal{P}}$
  (altrimenti, si potrebbe procedere con $\overline{L}_{\mathcal{P}}$).

Dimostrazione del Teorema di Rice

Mostriamo $A_{TM} \leq_m L_{\mathcal{P}}$

- Sia $T_\emptyset$ una TM tale che $L(T_\emptyset) = \emptyset$;
- Possiamo assumere $\langle T_\emptyset \rangle \notin L_{\mathcal{P}}$
  (altrimenti, si potrebbe procedere con $\overline{L}_{\mathcal{P}}$).
- Poichè $L_{\mathcal{P}}$ è non banale, esiste una TM T tale che $\langle T \rangle \in L_{\mathcal{P}}$.

Dimostrazione del Teorema di Rice

Progettiamo una funzione f come segue

$$f(\langle M, w \rangle) = \langle S \rangle$$

dove S su input x:

Dimostrazione del Teorema di Rice

Progettiamo una funzione f come segue

$$f(\langle M, w\rangle) = \langle S\rangle$$

dove S su input x:

- Simula M con input w
  - Se M si ferma e rifiuta, allora S rifiuta x
  - Se M accetta, allora S simula T su input x:
    se T accetta x allora S accetta, se T rifiuta x allora S rifiuta.

Dimostrazione del Teorema di Rice

Progettiamo una funzione f come segue

$$f(\langle M, w\rangle) = \langle S\rangle$$

dove S su input x:

- Simula M con input w
  - Se M si ferma e rifiuta, allora S rifiuta x
  - Se M accetta, allora S simula T su input x:
    se T accetta x allora S accetta, se T rifiuta x allora S rifiuta.

Mostriamo che la funzione f é una riduzione

- f é calcolabile

Dimostrazione del Teorema di Rice

Progettiamo una funzione f come segue

$$f(\langle M, w\rangle) = \langle S\rangle$$

dove S su input x:

- Simula M con input w
  - Se M si ferma e rifiuta, allora S rifiuta x
  - Se M accetta, allora S simula T su input x:
    se T accetta x allora S accetta, se T rifiuta x allora S rifiuta.

Mostriamo che la funzione f é una riduzione

- f é calcolabile
- $\langle M, w\rangle \in A_{TM} \ \Rightarrow\ L(S) = L(T) \ \Rightarrow S \in L_{\mathcal{P}}$

Dimostrazione del Teorema di Rice

Progettiamo una funzione f come segue

$$f(\langle M, w \rangle) = \langle S \rangle$$

dove S su input x:

- Simula M con input w
  - Se M si ferma e rifiuta, allora S rifiuta x
  - Se M accetta, allora S simula T su input x:
    se T accetta x allora S accetta, se T rifiuta x allora S rifiuta.

Mostriamo che la funzione f é una riduzione

- f é calcolabile
- $\langle M, w \rangle \in A_{TM} \;\Rightarrow\; L(S) = L(T) \;\Rightarrow S \in L_{\mathcal{P}}$
- $\langle M, w \rangle \notin A_{TM} \;\Rightarrow\; L(S) = \emptyset = L(T_\emptyset) \;\Rightarrow S \notin L_{\mathcal{P}}$

Conseguenze del Teorema di Rice

Non possiamo decidere se una MdT:

- Accetta $\emptyset$.
- Accetta un linguaggio finito.
- Accetta un linguaggio regolare, ecc.