8 Decidibilità secondo turing

giovedì 21 novembre 2024 17:02

100-decidibilita-10.pdf

Automata, Languages and Computing

Decidibilità secondo Turing

1

1

Obiettivo

- dimostrare che alcuni problemi sono algoritmicamente risolvibili ed altri no
  - del primo fatto abbiamo ampia evidenza dalla esperienza di informatici

2

2

100-decidibilita-10.pdf

Problemi indecidibili

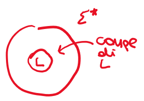
- un linguaggio L è *Turing-indecidibile* (o più semplicemente *indecidibile*) quando non esiste una MT che lo decida
  - cioè che accetti tutte le stringhe di L e rifiuti tutte le stringhe di Σ^* non appartenenti a L
- analoghe definizioni si possono dare per
  - la risolvibilità di problemi
  - la calcolabilità di funzioni

3

3

Una mt decide un linguaggio quando qualunque stringa di sgma asteriscato la mt fa una computazione massimale e la computazione si ferma in uno stato finale se la stringa è del linguaggio e si ferma in uno stato non finale se la stringa non è del linguaggio.
Però le mt possono anche riconoscere i linguaggi:
Riconosce L se proposta una stringa di sigma asteriscato se la computazione è massimale e si ferma su uno stato di F, altrimenti la stringa potrebbe essere rifiutata oppure ci possa essere una computazione infinita.
Un linguaggio indecidibile: la mt che esamina il linguaggio ci sta e potrebbe darsi che lo riconosca ma non sia in grado di deciderlo.
Mt che decidono e mt che riconoscono: se lo decide lo riconosce, se lo riconosce non necessariamente lo decide.

Il complemento di una mt che decide un linguaggio è ancora una mt che decide il complemento? Si.
I linguaggi decidibili sono chiusi rispetto alla complementazione: se posso decidere un linguaggio con una mt, posso decidere anche il suo complemento.



Stiamo parlando di mt deterministiche.

Il problema della fermata

- vogliamo realizzare una funzionalità di base per un debugger
- vogliamo scrivere del software che, dato un programma e dei dati di input, stabilisca se il programma termina su quei dati

È indecidibile questo problema

4

100-decidibilita-10.pdf

Un primo problema indecidibile

- molto simile al problema della fermata
- problema A_{TM} (linguaggio A_{TM})
  $A_{TM} = \{\langle M,w\rangle \mid$ M è una MT che accetta la stringa $w\}$
- in primo luogo osserviamo che A_{TM} è Turing-riconoscibile
  - sottoponiamo M e w alla MT universale U
  - se prima o poi U arriva ad uno stato accettante, allora accettiamo la stringa

5

Problema ATM è simile, ma diverso.

--> dobbiamo decidere se , data una mt M e la stringa w (sottoforma di coppia), la M accetta w?
Noi vogliamo decidere se questo è vero.
Io posso dire che questo è un L: linguaggio fatto da una coppia di stringhe, la prima è la codifica della mt e la seconda stringa è un input per la mt. Una stringa (la coppia) appartiene al linguaggio atm quando m corrisponde ad una mt che una volta eseguita accetta w. Il problema di decidere se M accetta W è uguale a stabilire se la stringa "Mw" appartiene a questo linguaggio
Altrimenti la stringa non appartiene al linguaggio.
Il linguaggio Atm è turing riconoscibile: esiste una mt che data una stringa se essa appartiene ad L risponde di si, altrimenti se non appartiene non so dire molto.

Prendo M, la sua codifica, la stringa w e la do in pasto alla mt U. U si comporta come M quando gli propongo w. Quando U si ferma mi chiedo se la stringa è stata accettata--> la stringa w fa parte di Atm, altrimenti se è stata rifiutata non ne fa parte. Ma la computazione può anche durare un tempo infinito.
-> Atm è turing riconoscibile. Ma anche decidibile? NO. Vedi dim sotto

Il problema A_{TM} è indecidibile

- teorema:
  il linguaggio $A_{TM} = \{\langle M,w\rangle \mid$ M è una MT che accetta la stringa $w\}$ è indecidibile
- dimostrazione:
  per assurdo, supponiamo che esista una macchina che decide A_{TM} ed otteniamo una contraddizione

6

Decidibilità secondo...

Registrazione audio avviata: 17:46 giovedì 21 novembre 2024

100-decidibilita-10.pdf

A_{TM} è indecidibile – dimostrazione

- sia H una MT che decide A_{TM}
  $H(\langle M,w\rangle)$ = accetta se M accetta w
  = rifiuta se M non accetta w
- costruiamo un'altra MT H' che usa H come subroutine
- H' usa H per stabilire cosa fa M quando riceve come input M stessa
- H' valuta $H(\langle M,M\rangle)$

7

Costruiamo H' con H.
H' valuta H(<M,M>): la uso come stringa

A_{TM} è indecidibile – dimostrazione

C esiste? Si, vista nelle lez precedenti.

- se H esiste, allora esiste anche H', infatti H' deve solo copiare M (usa una macchina C che fa la copia) e far partire H
- H'(M) = accetta se M accetta M
  = rifiuta se M non accetta M
- costruiamo ora un'altra macchina D che usa H' come subroutine
- D fa partire H' e restituisce l'opposto del risultato: se H' restituisce "accetta" D restituisce "rifiuta" e viceversa
- se H' esiste, allora esiste anche D (usa una macchina E che calcola l'opposto)

8

8

Siccome H' come prima cosa invoca C e poi H, allora H' esiste per l'esistenza di H e C.

D usa H' come strumento di lavoro. D sta per "diagonale".
H' esiste, H' decide e quando di ferma accetta o rifiuta. Quindi posso farne il complemento.

A_{TM} è indecidibile – dimostrazione

riassumendo:

- H(<M,w>) = accetta se M accetta w
  = rifiuta se M non accetta w
- H'(M) = accetta se M accetta M
  = rifiuta se M non accetta M
- D(M) = accetta se M non accetta M
  = rifiuta se M accetta M

9

9

A_{TM} è indecidibile – dimostrazione

- cosa succede se a D diamo in input D?

10

10

A_{TM} è indecidibile – dimostrazione

- H(<M,w>) = accetta se M accetta w
  = rifiuta se M non accetta w
- H'(M) = accetta se M accetta M
  = rifiuta se M non accetta M
- D(M) = accetta se M non accetta M
  = rifiuta se M accetta M
- D(D) = accetta se D non accetta D
  = rifiuta se D accetta D

-> assurdo

11

Decidibilità secondo...

Registrazione audio avviata: 18:05 giovedì 21 novembre 2024

11

A_{TM} è indecidibile – dimostrazione

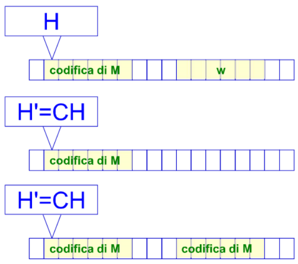
- qualunque cosa faccia D in realtà fa l'opposto: abbiamo un assurdo

12

12

100-decidibilita-10.pdf

Rivediamolo rappresentando il nastro



13

13

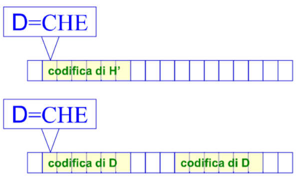
Decidibilità secondo...

Registrazione audio avviata: 18:06 giovedì 21 novembre 2024

Separata M e w da 3 spazi bianchi.

Copia + esecuzione di H

Rivediamolo rappresentando il nastro



D(D) = accetta se D non accetta D
= rifiuta se D accetta D

14

14

D: fa C, H e poi l'opposto

->D riceve in pasto se stesso

100-decidibilita-10.pdf

Decidibilità secondo...

Indecidibilità di A_{TM} e diagonalizzazione

- elenchiamo tutte le MT sulle righe e sulle colonne di una tabella
- la posizione i,j è *accetta* se, quando forniamo in input M_i a M_j, M_j accetta M_i ed è uno spazio se M_j

Registrazione audio avviata: 18:08 giovedì 21 novembre 2024

L'insieme dei numeri reali non è numerabile. Lo abbiamo dimostrato prendendo i reali in (0,1). Supponevamo esistesse la numerazione, ma comunque sia fatta la numerazione almeno un numero non ci sta. Quindi ci sta un numero che appartiene a (0,1) ma che non sta nell'enumerazione. Allora i reali non sono numerabili.

Vediamo le analogie con le mt.

rigetta o cicla

	M_1	M_2	M_3	…
M_1	*accetta*		*accetta*	…
M_2	*accetta*	*accetta*	*accetta*	…
M_3	*accetta*			…
….	….			…

15

15

Le mt sono infinite, ma sono numerabili e allora posso farmi un elenco delle mt fatte come mi pare su righe e colonne. Ho fatto una matrice bidimensionale infinita in righe e colonne.

Indecidibilità di A_{TM} e diagonalizzazione

- nella tabella che segue è mostrato il comportamento di H quando ha in input gli elementi della tabella precedente

	M_1	M_2	M_3	…
M_1	*accetta*	*rifiuta*	*accetta*	…
M_2	*accetta*	*accetta*	*accetta*	…
M_3	*accetta*	*rifiuta*	*rifiuta*	…
….	….			…

16

16

Ma la mt D dove la metto?

Indecidibilità di A_{TM} e diagonalizzazione

- nella tabella possiamo aggiungere D, che essendo una MT, prima o poi appare nell'elenco

	M_1	M_2	M_3	…	D	…
M_1	*accetta*	*rifiuta*	*accetta*	…	*accetta*	
M_2	*accetta*	*accetta*	*accetta*	…	*rifiuta*	
M_3	*accetta*	*rifiuta*	*rifiuta*	…	*accetta*	
….	….			…		
D	*rifiuta*	*rifiuta*	*accetta*	…		
….	….			…		

17

17

Indecidibilità di A_{TM} e diagonalizzazione

- ma D calcola esattamente l'opposto di quanto appaia sulla diagonale
- l'elemento in posizione D,D deve essere l'opposto di se stesso

	M_1	M_2	M_3	…	D	…
M_1	*accetta*	*rifiuta*	*accetta*	…	*accetta*	
M_2	*accetta*	*accetta*	*accetta*	…	*rifiuta*	
M_3	*accetta*	*rifiuta*	*rifiuta*	…	*accetta*	
….	….			…		
D	*rifiuta*	*rifiuta*	*accetta*	…	?	
….	….			…		

18

18

Guardo la riga D: nel punto in cui D(D) non sappiamo cosa scrivere perché qualunque cosa dovremmo scriverci ci dobbiamo scrivere l'opposto.

Calcolabilità secondo Turing

calcolabilità secondo Turing in vari contesti

- decisione di predicati
  - un *predicato* su Σ^* è una funzione $p:(\Sigma^*)^n \rightarrow \{vero, falso\}$
- un predicato p è *Turing-decidibile* se esiste una MT che calcola p, altrimenti è *Turing-indecidibile*
  - il predicato A_{TM} è Turing-indecidibile
- un predicato p è *semi-decidibile* se, pur essendo indecidibile, è *Turing-riconoscibile*
  - il predicato A_{TM} è semi-decidibile

19

19

Predicato: prendi una funzione con argomenti e in base agli argomenti rispondi vero o falso

Riconoscere Atm è fattibile -> con una mt universale. Decidere Atm non lo è

Un linguaggio non Turing-riconoscibile

- pur essendo indecidibile, il linguaggio A_{TM} è riconoscibile (è semi-decidibile)
- esistono linguaggi che non sono neppure riconoscibili
- diciamo che un linguaggio è co-riconoscibile se il suo complemento è riconoscibile

20

20

Il complemento di Atm è riconoscibile? Il predicato complemento di Atm è semi-decidibile? Se il complemento di atm fosse riconoscibile allora avremmo che atm è decidibile. Ma non è così, e allora il complemento di Atm è non riconoscibile.

100-decidibilita-10.pdf

Un linguaggio non Turing-riconoscibile

- teorema:
  un linguaggio è decidibile se e solo se è sia Turing-riconoscibile sia co-Turing-riconoscibile
- dimostrazione ($\Rightarrow$):
  - consideriamo il linguaggio A e supponiamo sia decidibile
  - se A è decidibile per definizione è anche Turing-riconoscibile
  - se A è decidibile il suo complemento è decidibile, cioè A è co-Turing-riconoscibile
  - ciò completa la prima parte della dimostrazione

21

21

Un linguaggio non Turing-riconoscibile

- dimostrazione ($\Leftarrow$):
  - A e $\underline{A}$ sono entrambi Turing-riconoscibili
  - siano M ed N le MT che riconoscono A e $\underline{A}$
  - costruiamo una MT che decide A eseguendo M ed N in *parallelo* sullo stesso input
  - se M accetta la macchina accetta, se N accetta la macchina rifiuta

22

22

100-decidibilita-10.pdf

Un linguaggio non Turing-riconoscibile

- teorema:
  $\underline{A}_{TM}$ non è Turing-riconoscibile
- dimostrazione:
  - se lo fosse, A_{TM} sarebbe decidibile

23

23