L'enumerabilità ricorsiva (cenni)

Eugenio G. Omodeo

UNIVERSITÀ
DEGLI STUDI DI TRIESTE

Dip. Matematica e Geoscienze — DMI

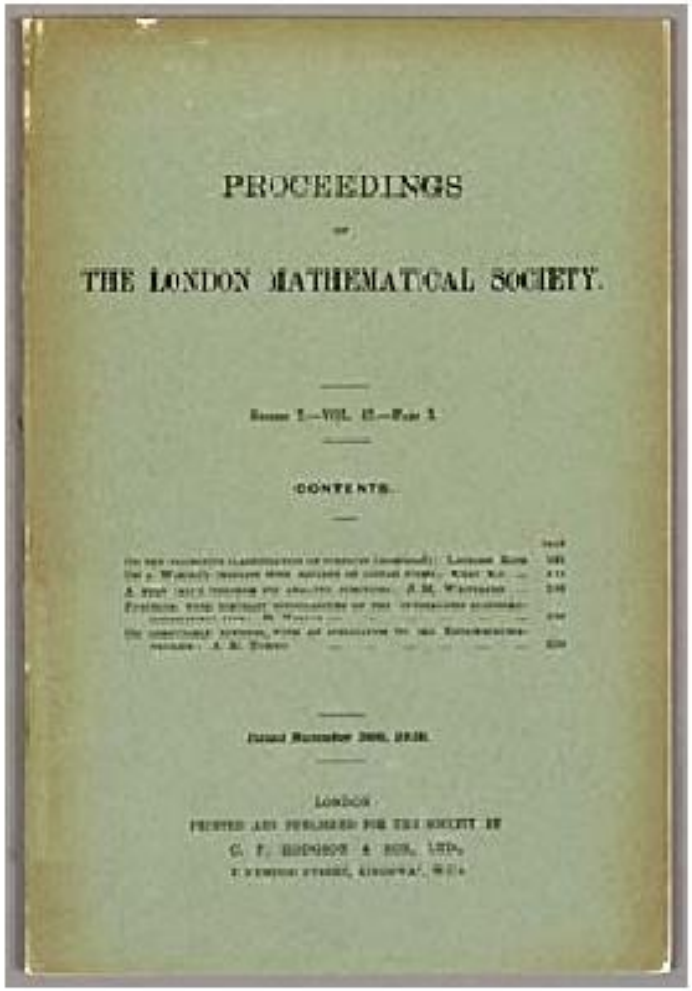
Trieste, 23/03/2016

Esplicitazione del concetto di *calcolabilità*

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. TURING

[Received 28 May, 1936.—Read 12 November, 1936.]


PROCEEDINGS

OF

THE LONDON MATHEMATICAL SOCIETY.

CONTENTS.

Computazioni sui naturali

Sia π un *programma*[1] la cui esecuzione

0.

1.

2.

1.

2.

[1] In un linguaggio *Turing-completo*, i.e., a versatilità completa

Computazioni sui naturali

Sia π un *programma*[1] la cui esecuzione

0. ricevendo come dato di partenza un qualsiasi $x \in \mathbb{N}$ ...
1.
2.

1.
2.

[1] In un linguaggio *Turing-completo*, i.e., a versatilità completa

Computazioni sui naturali

Sia π un *programma*[1] la cui esecuzione

0. ricevendo come dato di partenza un qualsiasi $x \in \mathbb{N}$ . . .
1. . . . giunga a termine, fornendo un risultato $y \in \mathbb{N}$; oppure
2.

1.
2.

[1] In un linguaggio *Turing-completo*, i.e., a versatilità completa

Computazioni sui naturali

Sia π un *programma*[1] la cui esecuzione

0. ricevendo come dato di partenza un qualsiasi $x \in \mathbb{N}$ . . .
1. . . . giunga a termine, fornendo un risultato $y \in \mathbb{N}$; oppure
2. . . . si protragga per sempre, nel qual caso il risultato manca

1.
2.

[1] In un linguaggio *Turing-completo*, i.e., a versatilità completa

Computazioni sui naturali

Sia π un *programma*[1] la cui esecuzione

0. ricevendo come dato di partenza un qualsiasi $x \in \mathbb{N}$ ...
1. ...giunga a termine, fornendo un risultato $y \in \mathbb{N}$; oppure
2. ...si protragga per sempre, nel qual caso il risultato manca

Nelle rispettive situazioni scriviamo:

1.
2.

[1] In un linguaggio *Turing-completo*, i.e., a versatilità completa

Computazioni sui naturali

Sia π un *programma*[1] la cui esecuzione

0. ricevendo come dato di partenza un qualsiasi $x \in \mathbb{N}$ ...
1. ...giunga a termine, fornendo un risultato $y \in \mathbb{N}$; oppure
2. ...si protragga per sempre, nel qual caso il risultato manca

Nelle rispettive situazioni scriviamo:

1. $\psi\pi(x) = y$
2.

[1] In un linguaggio *Turing-completo*, i.e., a versatilità completa

Sia π un *programma*[1] la cui esecuzione

0. ricevendo come dato di partenza un qualsiasi $x \in \mathbb{N}$ ...
1. ...giunga a termine, fornendo un risultato $y \in \mathbb{N}$; oppure
2. ...si protragga per sempre, nel qual caso il risultato manca

Nelle rispettive situazioni scriviamo:

1. $\psi_\pi(x) = y$
2. $\psi_\pi(x) = \perp$

[1] In un linguaggio *Turing-completo*, i.e., a versatilità completa

Esempio: Valutazione di un polinomio

Se $P \in \mathbb{Z}[x]$, possiamo implementare un programma π tale che per ogni $x \in \mathbb{N}$,

$$x \overset{\psi_\pi}{\longmapsto} \begin{cases} P(x) & \text{quando} \quad P(x) \geqslant 0 \\ \bot & \text{altrim.} \end{cases}$$

Ogni funzione *parziale*

$$\psi_\pi : \mathbb{N} \rightharpoonup \mathbb{N}$$

associata c.s. a un programma π vien detta *computabile*

Ogni funzione *parziale*

$$\psi_\pi : \mathbb{N} \rightharpoonup \mathbb{N}$$

associata c.s. a un programma π vien detta *computabile*

Se inoltre ψ_π è *iniettiva* e, per ogni $x \in \mathbb{N}$,

$$\psi_\pi(x+1) \neq \perp \implies \psi_\pi(x) \neq \perp \ ,$$

allora diciamo che ψ_π è un *elenco*

Si chiama:

1. [ ] ogni insieme che sia *dominio*,

$$\{x \in \mathbb{N} \mid \psi\pi(x) \neq \perp\},$$

2. [ ]

Si chiama:

1. *enumerabile ricorsivamente* ogni insieme che sia *dominio*,

$$\{x \in \mathbb{N} \mid \psi\pi(x) \neq \perp\},$$

di una funzione parziale computabile;

2.

Si chiama:

1. *enumerabile ricorsivamente* ogni insieme che sia *dominio*,

$$\{x \in \mathbb{N} \mid \psi\pi(x) \neq \perp\},$$

di una funzione parziale computabile;

2. ogni *insieme-immagine*

$$\{\psi\pi(0), \psi\pi(1), \psi\pi(2), \ldots\} \setminus \{\perp\},$$

Si chiama:

1. *enumerabile ricorsivamente* ogni insieme che sia *dominio*,

$$\{x \in \mathbb{N} \mid \psi\pi(x) \neq \perp\},$$

di una funzione parziale computabile;

2. *elencabile* ogni *insieme-immagine*

$$\{\psi\pi(0), \psi\pi(1), \psi\pi(2), \ldots\} \setminus \{\perp\},$$

di un elenco

Si chiama:

1. *enumerabile ricorsivamente* ogni insieme che sia *dominio*,

$$\{x \in \mathbb{N} \mid \psi\pi(x) \neq \perp\},$$

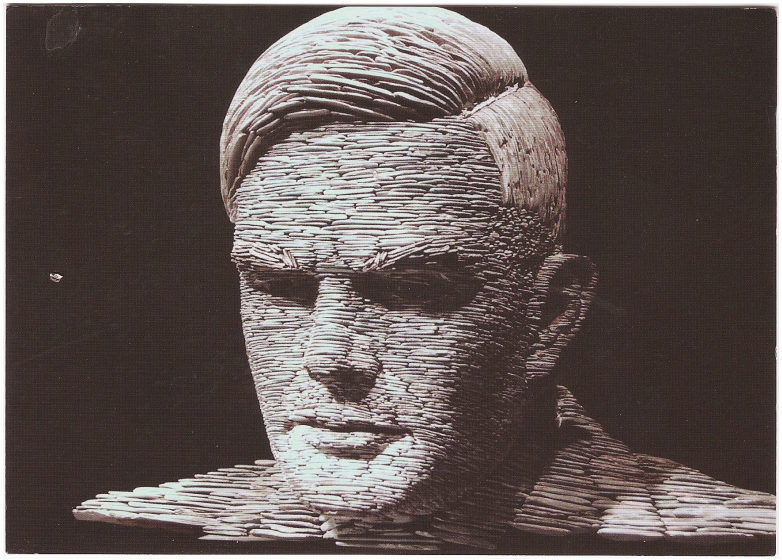
di una funzione parziale computabile;

2. *elencabile* (per quanto alla rinfusa) ogni *insieme-immagine*

$$\{\psi\pi(0), \psi\pi(1), \psi\pi(2), \ldots\} \setminus \{\perp\},$$

di un elenco

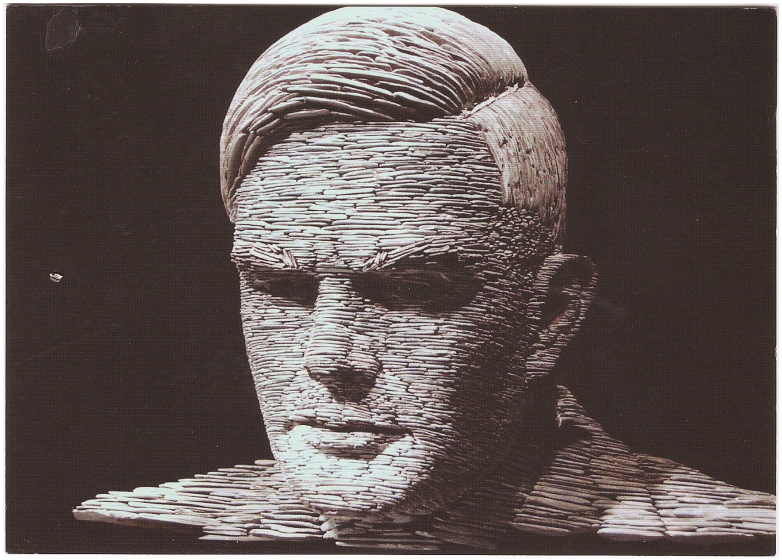
Conseguenze della completezza di Turing



Alan Mathison Turing (1912-1954)

- Le nozioni di insieme
  - enumerabile ricorsivamente (r.e.)
  - elencabile

Conseguenze della completezza di Turing

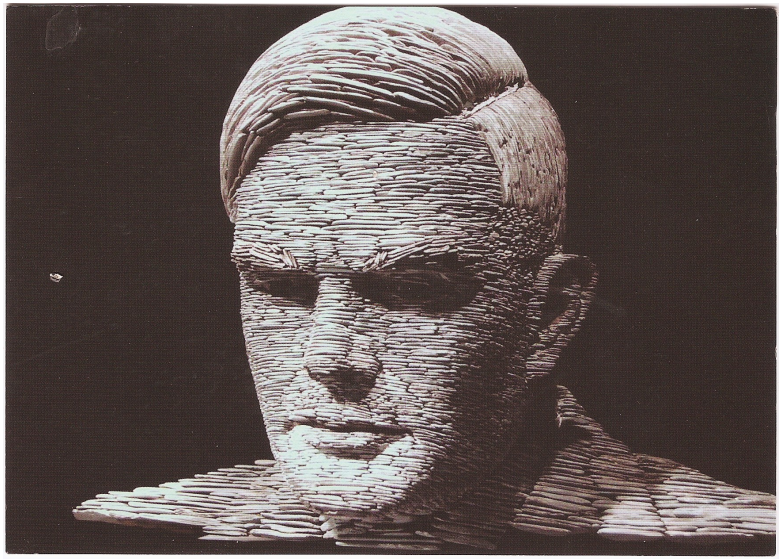


Alan Mathison Turing (1912-1954)

- Le nozioni di insieme
  - enumerabile ricorsivamente (r.e.)
  - elencabile

  si equivalgono !

Conseguenze della completezza di Turing



Alan Mathison Turing (1912-1954)

- Le nozioni di insieme
  - enumerabile ricorsivamente (r.e.)
  - elencabile

  si equivalgono !

- Esistono insiemi D r.e. per i quali $\mathbb{N} \setminus D$ *non è* r.e.

Semidecidibilità e Decidibilità

- Quando D è r.e., allora possiamo

Semidecidibilità e Decidibilità

- Quando D è r.e., allora possiamo

  accertare, di ogni suo elemento x, **che** davvero gli appartiene

-

Semidecidibilità e Decidibilità

- Quando D è r.e., allora possiamo

  accertare, di ogni suo elemento x, **che** davvero gli appartiene

- Se D ed $\mathbb{N} \setminus D$ sono entrambi r.e. allora, interfogliandone gli elenchi, possiamo

Semidecidibilità e Decidibilità

- Quando D è r.e., allora possiamo

  accertare, di ogni suo elemento x, **che** davvero gli appartiene

- Se D ed $\mathbb{N} \setminus D$ sono entrambi r.e. allora, interfogliandone gli elenchi, possiamo

  stabilire, di ogni x in $\mathbb{N}$, **se** appartenga o meno a D

5. *Il contributo di Raphael.*

Raphael Robinson era interessato a ciò che avevo dimostrato nella mia dissertazione, vale a dire che ogni insieme elencabile può essere definito nel modo che segue:

$$(\exists y)\,(\forall k \leq y)\,(\exists x_1, x_2, \ldots, x_n)[p(a, k, y, x_1, x_2, \ldots, x_n) = 0],$$

dove p è un polinomio. La chiamò *forma normale di Davis* e si domandò quanto potesse essere piccolo il numero n. Riuscí a dimostrare che 4 era abbastanza grande, ovvero che ogni insieme elencabile può essere rappresentato come

$$(\exists y)\,(\forall k \leq y)\,(\exists u, v, w, x)[p(a, k, y, u, v, w, x) = 0].$$

Di nuovo, mi chiesero di fare da *referee*, cosí studiai la dimostrazione con molta cura. Era complicata e difficile, un vero tour de force che rivelava il talento matematico di Raphael.[10]

[10] Molto tempo dopo continuò a interessarsi a questa forma normale e ridusse il numero dei quantificatori esistenziali da quattro a tre. Matijasevič mostrò in seguito che anche due quantificatori esistenziali sono sufficienti.

[Dav10]

Esercizio. Individuare una una schiera $K_n(a_1, \ldots, a_n)$ di polinomi a coefficienti in $\mathbb{Z}$ che per tutti gli $n \in \mathbb{N}$ soddisfino le condizioni:

1. $0 \leqslant \boldsymbol{a}_i \leqslant K_n(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n)$ per ogni $\langle \boldsymbol{a}_1, \ldots, \boldsymbol{a}_n \rangle \in \mathbb{N}^n$ ed $i = 1, \ldots, n$;
2. iniettività della funzione $\vec{\boldsymbol{a}} \mapsto K_n(\vec{\boldsymbol{a}})$ per $\vec{\boldsymbol{a}} \in \mathbb{N}^n$ $\dashv$

Esercizio. Individuare una una schiera $K_n(a_1, \ldots, a_n)$ di polinomi a coefficienti in $\mathbb{Z}$ che per tutti gli $n \in \mathbb{N}$ soddisfino le condizioni:

1. $0 \leqslant a_i \leqslant K_n(a_1, \ldots, a_n)$ per ogni $\langle a_1, \ldots, a_n \rangle \in \mathbb{N}^n$ ed $i = 1, \ldots, n$;
2. iniettività della funzione $\vec{a} \mapsto K_n(\vec{a})$ per $\vec{a} \in \mathbb{N}^n$ $\dashv$

A che scopo? Potremmo caratterizzare gli r.e. n-dimensionali come quegli $\mathcal{R} \subseteq \mathbb{N}^n$ per i quali

$$\{ K_n(\vec{a}) : \vec{a} \in \mathcal{R} \}$$

è un insieme r.e.

Esplicitazione del concetto di *universalità*

"It is possible to invent a single machine which can be used to compute any computable sequence. If this machine I is supplied with a tape on the beginning of which is written the $S.D$ of some computing machine M, then I will compute the same sequence as M. In this section I explain in outline the behavior of the machine. The next section is devoted to giving the complete table for I." **Alan Mathison Turing, 1936**

Teorema (Turing, 1936)

Per ogni n si può esibire un programma $\mathcal{U}_n$ tale che, indicate con

$$\psi_\pi^{(n)}(x_1, \ldots, x_n) \qquad \text{e con}$$

$$\psi_{\mathcal{U}_n}^{(n+1)}(x_1, \ldots, x_n, x_{n+1})$$

- la funzione n-aria computata da un programma π
- la funzione $n+1$-aria computata da $\mathcal{U}_n$

e con $\#\pi$ il numero di Gödel che compete a π, risulti

$$\psi_\pi^{(n)}(x_1, \ldots, x_n) \quad = \quad \psi_{\mathcal{U}_n}^{(n+1)}(x_1, \ldots, x_n, \#\pi)$$

per ogni $\langle x_1, \ldots, x_n \rangle \in \mathbb{N}^n$ ed ogni π

Voci bibliografiche

Martin Davis.

Il decimo problema di Hilbert: equazioni e computabilità.

In Claudio Bartocci and Piergiorgio Odifreddi, editors, *La matematica – Pensare il mondo,* Volume IV, Grandi Opere. Einaudi, 2010.

Martin D. Davis, Ron Sigal, and Elaine J. Weyuker.

Computability, complexity, and languages - Fundamentals of theoretical computer science.

Computer Science ad scientific computing. Academic Press, 1994.

A. M. Turing.

On Computable Numbers, with an application to the Entscheidungsproblem.

Proc. London Math. Soc., 2(42):230–265, 1936.

Correzione, ibid., (43):44-546, 1937.