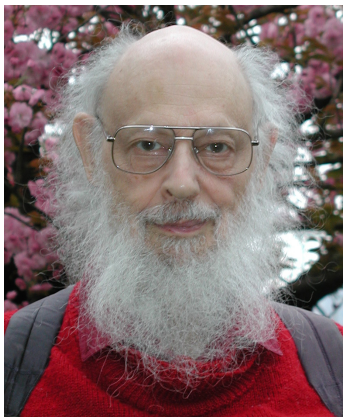
L'Omega di Chaitin nella Teoria Algoritmica dell'Informazione

Gianluca Basso

Università di Pisa

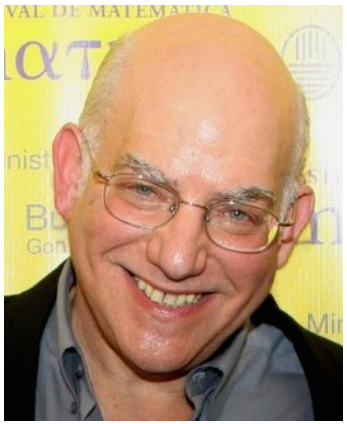
8 maggio 2014

Obiettivo



Ray Solomonoff (July 25, 1926 – December 7, 2009)

Andrey N. Kolmogorov (25 April 1903 – 20 October 1987)



Gregory J. Chaitin (born 15th. November, 1947)

Obiettivo

Introdurre la Teoria Algoritmica dell'Informazione (anche detta Teoria della Complessità di Kolmogorov) e alcuni risultati rilevanti. In particolare, presentare le proprietà dell'Ω di Chaitin.

Nozioni di base

Dato un alfabeto finito $A = \{a_1, \ldots, a_Q\},\ Q \geq 2$ di lettere $a_1, \ldots, a_Q$, denotiamo con A^* l'insieme di tutte le stringhe di lunghezza finita $x_1 x_2 \ldots x_n$, di lettere di A.

La lunghezza $|x|$ di una stringa $x = x_1 x_2 \ldots x_n$ è n.

Scriviamo λ per indicare la stringa nulla, ovvero la stringa di lunghezza 0.

Un ordinamento totale su A induce un buon ordinamento su A^* (ordine semi lessicografico).

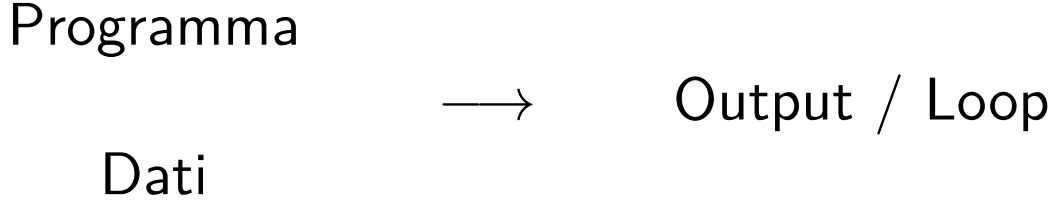
Cos'è un computer?

Cos'è un computer?

Programma

Dati

Cos'è un computer?



Computer di Chaitin

Computer di Chaitin

Un computer di Chaitin C è una funzione calcolabile parziale $C : A^* \times A^* \to A^*$ tale che, $\forall v \in A^*$, la funzione

$$\begin{aligned} c_v : & A^* \to A^* \\ & x \mapsto C(x, v) \end{aligned}$$

ha domino privo di prefissi.

Computer di Chaitin

Computer di Chaitin

Un computer di Chaitin C è una funzione calcolabile parziale $C : A^* \times A^* \to A^*$ tale che, $\forall v \in A^*$, la funzione

$$\begin{aligned} c_v : & A^* \to A^* \\ & x \mapsto C(x, v) \end{aligned}$$

ha domino privo di prefissi.

Un insieme S di stringhe di A^* è detto privo di prefissi se $\forall x, y \in S$

$$y = xz \Rightarrow z = \lambda \text{ (e quindi anche } y = x)$$

ovvero nessun elemento di S è prefisso di un altro elemento di S.

Computer di Chaitin Universale

Computer di Chaitin Universale

Un computer di Chaitin U è universale se, per ogni computer di Chaitin C, esiste una costante k tale che, se $(x, y) \in \mathrm{Dom}(C)$, esiste una stringa x' tale che

$$U(x', y) = C(x, y) \quad \text{e} \quad |x'| \leq |x| + k$$

Computer di Chaitin Universale

Computer di Chaitin Universale

Un computer di Chaitin U è universale se, per ogni computer di Chaitin C, esiste una costante k tale che, se $(x, y) \in \mathrm{Dom}(C)$, esiste una stringa x' tale che

$$U(x', y) = C(x, y) \quad \text{e} \quad |x'| \leq |x| + k$$

Esiste effettivamente un computer di Chaitin universale.

Programma canonico e informazione assoluta

Data una stringa $x \in A^*$, il programma canonico per computare x in U di Chaitin universale è

$$x^* = \min\{u \in A^* : U(u, \lambda) = x\}$$

dove il minimo è preso sull'ordine semi lessicografico su A^*.

Programma canonico e informazione assoluta

Data una stringa $x \in A^*$, il programma canonico per computare x in U di Chaitin universale è

$$x^* = \min\{u \in A^* : U(u, \lambda) = x\}$$

dove il minimo è preso sull'ordine semi lessicografico su A^*.

Informazione assoluta

Dato un computer di Chaitin universale U, l'informazione assoluta della stringa $x \in A^*$ è

$$I_U(x) = |x^*|$$

Programma canonico e informazione assoluta

Data una stringa $x \in A^*$, il programma canonico per computare x in U di Chaitin universale è

$$x^* = \min\{u \in A^* : U(u, \lambda) = x\}$$

dove il minimo è preso sull'ordine semi lessicografico su A^*.

Informazione assoluta

Dato un computer di Chaitin universale U, l'informazione assoluta della stringa $x \in A^*$ è

$$I_U(x) = |x^*|$$

L'informazione assoluta non è computabile.

L'insieme dei programmi canonici è immune.

Teorema

L'insieme dei programmi canonici $CP = \{x^* : x \in A^*\}$ rispetto ad un computer di Chaitin universale U è infinito e non ha sottoinsiemi infiniti che siano ricorsivamente enumerabili.

L'insieme dei programmi canonici è immune.

Teorema

L'insieme dei programmi canonici $CP = \{x^* : x \in A^*\}$ rispetto ad un computer di Chaitin universale U è infinito e non ha sottoinsiemi infiniti che siano ricorsivamente enumerabili.

Immediata conseguenza: la funzione

$$f_U : A^* \to A^*$$
$$x \mapsto x^*$$

non è computabile.

Omega di Chaitin

La probabilità algoritmica assoluta che U restituisca x è

$$P_U(x) = \sum_{\{u \in A^* : U(u,\lambda)=x\}} Q^{-|u|}$$

dove Q è il numero di lettere dell'alfabeto A.

Omega di Chaitin

La probabilità algoritmica assoluta che U restituisca x è

$$P_U(x) = \sum_{\{u \in A^* : U(u,\lambda)=x\}} Q^{-|u|}$$

dove Q è il numero di lettere dell'alfabeto A.

Omega di Chaitin

Dato un computer di Chaitin universale U, chiamiamo probabilità d'arresto di U la quantità:

$$\Omega_U = \sum_{x \in A^*} P_U(x)$$

Omega di Chaitin

La probabilità algoritmica assoluta che U restituisca x è

$$P_U(x) = \sum_{\{u \in A^* : U(u,\lambda)=x\}} Q^{-|u|}$$

dove Q è il numero di lettere dell'alfabeto A.

Omega di Chaitin

Dato un computer di Chaitin universale U, chiamiamo probabilità d'arresto di U la quantità:

$$\Omega_U = \sum_{x \in A^*} P_U(x) \leq 1$$

Randomness

Fissiamo un computer di Chaitin universale U. Scriviamo $I(x)$ in luogo di $I_U(x)$ e $P(x)$ in luogo di $P_U(x)$.

Stringa casuale

Una stringa $x \in A^n$ è Chaitin-casuale se

$$I(x) = \max_{u \in A^n} I(u)$$

Randomness

Fissiamo un computer di Chaitin universale U. Scriviamo $I(x)$ in luogo di $I_U(x)$ e $P(x)$ in luogo di $P_U(x)$.

Stringa casuale

Una stringa $x \in A^n$ è Chaitin-casuale se

$$I(x) = \max_{u \in A^n} I(u)$$

Grado di Casualità di Chaitin

Una stringa $x \in A^n$ ha grado di casualità di Chaitin m se

$$I(x) \geq \max_{y \in A^n} I(y) - m$$

Un esempio

Un esempio

000000000000000000000000

Un esempio

$$000000000000000000000000 \longrightarrow 0^{24}$$

Un esempio

$$000000000000000000000000 \longrightarrow 0^{24}$$

$$001001001001001001001001$$

Un esempio

$$000000000000000000000000 \longrightarrow 0^{24}$$

$$001001001001001001001001 \longrightarrow (001)^{8}$$

Un esempio

$$000000000000000000000000 \longrightarrow 0^{24}$$

$$001001001001001001001001 \longrightarrow (001)^{8}$$

$$000100100000000000100010$$

Un esempio

$000000000000000000000000 \longrightarrow 0^{24}$

$001001001001001001001001 \longrightarrow (001)^{8}$

$000100100000000000100010 \longrightarrow$ Ordino le stringhe di lunghezza 24 rispetto alla quantità di 1 e poi lessicograficamente. Al programma che stampa l'i-esima stringa do in pasto l'indice della stringa.

Un esempio

$000000000000000000000000 \longrightarrow 0^{24}$

$001001001001001001001001 \longrightarrow (001)^{8}$

$000100100000000000100010 \longrightarrow$ Ordino le stringhe di lunghezza 24 rispetto alla quantità di 1 e poi lessicograficamente. Al programma che stampa l'i-esima stringa do in pasto l'indice della stringa.

001110101011101010010001

Un esempio

$000000000000000000000000 \longrightarrow 0^{24}$

$001001001001001001001001 \longrightarrow (001)^{8}$

$000100100000000000100010 \longrightarrow$ Ordino le stringhe di lunghezza 24 rispetto alla quantità di 1 e poi lessicograficamente. Al programma che stampa l'i-esima stringa do in pasto l'indice della stringa.

$001110101011101010010001 \longrightarrow 001110101011101010010001$

Algorithmic Coding Theorem

Algorithmic Coding Theorem

$$I(x) = -log_Q P(x) + O(1)$$

Algorithmic Coding Theorem

Caratteri della *Divina Commedia*: 548276

Algorithmic Coding Theorem

Caratteri della *Divina Commedia*: 548276 $\longrightarrow$ 4386208 *bit*.

Algorithmic Coding Theorem

Caratteri della *Divina Commedia*: 548276 $\longrightarrow$ 4386208 *bit*.

La probabilità che una scimmia riscriva la *Divina Commedia* su una macchina da scrivere (con 2 tasti) è $2^{-4386208}$.

Algorithmic Coding Theorem

Caratteri della *Divina Commedia*: 548276 $\longrightarrow$ 4386208 *bit*.

La probabilità che una scimmia riscriva la *Divina Commedia* su una macchina da scrivere (con 2 tasti) è $2^{-4386208}$.

Algorithmic Coding Theorem

La probabilità che una scimmia riscriva la *Divina Commedia* su una macchina da scrivere (con 2 tasti) è $2^{-4386208}$.

$$I(\textit{Divina commedia}) = \frac{\textit{bit}}{\text{fattore di compressione}}$$

Algorithmic Coding Theorem

La probabilità che una scimmia riscriva la *Divina Commedia* su una macchina da scrivere (con 2 tasti) è $2^{-4386208}$.

$$I(\textit{Divina commedia}) = \frac{\textit{bit}}{\text{fattore di compressione}} \approx 1000000$$

Algorithmic Coding Theorem

La probabilità che una scimmia riscriva la *Divina Commedia* su una macchina da scrivere (con 2 tasti) è $2^{-4386208}$.

$$I(\textit{Divina commedia}) = \frac{\textit{bit}}{\text{fattore di compressione}} \approx 1000000$$

Quindi, per il teorema appena enunciato:
$P(\textit{Divina Commedia}\ ) \approx 2^{-I(D.C.)} = 2^{-1000000}$.

Algorithmic Coding Theorem

Ovvero, se la scimmia la mettiamo a battere su un computer è esponenzialmente più probabile che l'esperimento abbia esito positivo.

Successioni casuali

Successione casuale

Una successione $\mathbf{x} \in A^{\omega}$ è casuale se e solo se

$$\lim_{n \to \infty} (I(\mathbf{x}(n)) - n) = \infty$$

dove $\mathbf{x}(n)$ è la stringa ottenuta troncando la successione $\mathbf{x}$ all' n-esima entrata.

Successioni casuali

Successione casuale

Una successione $\mathbf{x} \in A^{\omega}$ è casuale se e solo se

$$\lim_{n \to \infty} (I(\mathbf{x}(n)) - n) = \infty$$

dove $\mathbf{x}(n)$ è la stringa ottenuta troncando la successione $\mathbf{x}$ all' n-esima entrata.

L'espansione binaria di Ω_U è una successione casuale.

Ω è normale

Calude - 1994

Per ogni computer di Chaitin universale U, Ω_U è un numero trascendente, non computabile e normale tra 0 e 1.

Cristian S. Calude (born 21 April 1952)

Numero Borel-normale

Definizione

Un numero reale x si dice normale se, per qualunque intero $b \geq 2$, nell'espansione di x in base b tutte le cifre occorrono con probabilità $\frac{1}{b}$, tutte le coppie di cifre occorrono con probabilità $\frac{1}{b^2}$, tutte le triple di cifre occorrono con probabilità $\frac{1}{b^3}$...

Numero Borel-normale

Definizione

Un numero reale x si dice normale se, per qualunque intero $b \geq 2$, nell'espansione di x in base b tutte le cifre occorrono con probabilità $\frac{1}{b}$, tutte le coppie di cifre occorrono con probabilità $\frac{1}{b^2}$, tutte le triple di cifre occorrono con probabilità $\frac{1}{b^3}$...

Borel (1909) ha dimostrato che quasi ogni reale (secondo la misura di Lebesgue) è normale, ma è un problema aperto se i seguenti reali sono normali:

$$\sqrt{2},\ \pi,\ \ln(2),\ e$$

Risultati di Indipendenza - Solovay

Robert M. Solovay (born December 15, 1938)

Solovay - 1999

Sia T una teoria ricorsivamente enumerabile e 1-coerente nella quale è interpretabile PA, allora c'è un computer di Chaitin U tale che

- $PA \vdash (U \text{ è universale})$
- T non determina neanche un bit dell'epansione binaria di Ω_U

Risultati di Indipendenza - Solovay

Robert M. Solovay (born December 15, 1938)

Solovay - 1999

Sia T una teoria ricorsivamente enumerabile e 1-coerente nella quale è interpretabile PA, allora c'è un computer di Chaitin U tale che

- $PA \vdash (U \text{ è universale})$
- T non determina neanche un bit dell'epansione binaria di Ω_U

In particolare, se ZFC è 1-coerente allora non determina alcun bit di Ω_U.

Ω è un oracolo per il problema della fermata.

Chaitin - 1975

Conoscendo i primi n elementi dell'espansione binaria di Ω_U, è possibile decidere se $U(x, \lambda) < \infty$ per ogni stringa x di lunghezza al più n.

Ω è un oracolo per il problema della fermata.

Chaitin - 1975

Conoscendo i primi n elementi dell'espansione binaria di Ω_U, è possibile decidere se $U(x, \lambda) < \infty$ per ogni stringa x di lunghezza al più n.

Ma l'espansione binaria di Ω non è computabile (e non la conosciamo).

Ω è un oracolo per il problema della fermata.

Chaitin - 1975

Conoscendo i primi n elementi dell'espansione binaria di Ω_U, è possibile decidere se $U(x, \lambda) < \infty$ per ogni stringa x di lunghezza al più n.

Ma l'espansione binaria di Ω non è computabile (e non la conosciamo).

Anche ogni enunciato finitamente refutabile sarebbe deciso, avendo Ω come oracolo: a.e. Congettura di Goldbach.

Ω è un oracolo per il problema della fermata.

Dimostrazione:

$$\Omega_U = \sum_{y \in A_2^*} \sum_{\{u \in A_2^* : U(u,\lambda)=y\}} 2^{-|u|} = 0.\Omega_1\Omega_2 \ldots \Omega_n \ldots$$

Ω è un oracolo per il problema della fermata.

Dimostrazione:

$$\Omega_U = \sum_{y \in A_2^*} \sum_{\{u \in A_2^* : U(u,\lambda)=y\}} 2^{-|u|} = 0.\Omega_1\Omega_2 \ldots \Omega_n \ldots$$

Ora, qualsiasi computazione che termina, ovvero tale che $U(u_i, \lambda) < \infty$, contribuisce per $2^{-|u_i|}$ all'approssimazione di Ω_U. Dopo un tempo finito, otterremo una approssimazione di Ω_U che sia migliore di $0.\Omega_1\Omega_2 \ldots \Omega_n$. Possiamo allora dire che se x non è tra i programmi che hanno terminato la computazione allora $U(x, \lambda) = \infty$. Infatti, se x fosse tale che $U(x, \lambda) = y$ per un certo y, allora all'approssimazione di Ω_U verrebbe sommato $2^{-|x|} \geq 2^{-n}$, ottenendo una variazione nelle prime n posizioni. Poiché queste sono date, otteniamo un assurdo.

Grazie per l'attenzione.

Bibliografia


Cristian S. Calude.

Information and randomness.

Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, second edition, 2002.

An algorithmic perspective, With forewords by Gregory J. Chaitin and Arto Salomaa.


Gregory J. Chaitin.

A theory of program size formally identical to information theory.

J. Assoc. Comput. Mach., 22:329–340, 1975.


Thomas M. Cover and Joy A. Thomas.

Elements of information theory.

Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, second edition, 2006.


Robert M. Solovay.

A version of Ω for which ZFC cannot predict a single bit.

In *Finite versus infinite*, Discrete Math. Theor. Comput. Sci. (Lond.), pages 323–334. Springer, London, 2000.