Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Matematica

STOCASTICITÀ E INCOMPRIMIBILITÀ ALGORITMICA

Tesi di Laurea in Teoria dell'Informazione

Relatore:
Chiar.mo Prof.
Simone Martini

Presentata da:
Luca Wehrstedt

II Sessione
Anno Accademico 2013/14

Introduzione

L'argomento di questa tesi è a cavallo tra matematica e informatica teorica e giace nell'intersezione tra la teoria della probabilità e la teoria algoritmica dell'informazione. Quest'ultima è un ambito di ricerca recente (fondato nel 1965 da Andrei Kolmogorov), frutto dell'impianto delle idee della teoria della calcolabilità nella teoria dell'informazione classica, basata sull'approccio probabilistico di Shannon.

Partiremo dal lavoro originale di Kolmogorov per definire una misura della quantità di informazione contenuta in una stringa tramite un approccio computazionale: la complessità di una stringa sarà la lunghezza del più corto programma capace di produrla. Vedremo poi gli sviluppi moderni di questa teoria, in particolare i contributi di Chaitin, e noteremo subito i forti legami con la teoria della probabilità e con l'entropia di Shannon. Successivamente proporremo di identificare le stringhe casuali (nel senso intuitivo) con quelle algoritmicamente incomprimibili, seguendo l'idea che minore comprimibilità significhi minore regolarità e dunque maggiore casualità. Infine vedremo che, in effetti, le stringhe incomprimibili soddisfano tutte le proprietà stocastiche effettivamente verificabili, cioè le proprietà che la teoria della probabilità attribuisce a successioni di variabili aleatorie indipendenti e identicamente distribuite. Faremo ciò in maniera generale utilizzando la notevole teoria di Martin-Löf e poi vedremo in dettaglio l'aspetto della normalità di Borel.

Indice

Capitolo 1

Preliminari

Assumeremo che il lettore abbia dimestichezza con la teoria della calcolabilità: lavoreremo molto con funzioni calcolabili, anche se non faremo ricorso a risultati particolarmente avanzati. Utilizzeremo anche concetti della teoria della probabilità, impostati secondo l'approccio assiomatico della teoria della misura. Infine risulterà comodo (ma non indispensabile) conoscere la teoria dell'informazione classica, soprattutto definizioni e formule notevoli dell'entropia di Shannon e delle quantità ad essa relate (un eventuale testo di riferimento può essere [3]).

Nel seguito richiameremo alcuni risultati, introdurremo delle definizioni utili per il resto della trattazione e stabiliremo delle convenzioni di notazione.

Stringhe

In tutta la trattazione denoteremo con Σ l'alfabeto binario, cioè $\Sigma = \{0, 1\}$. Chiameremo Σ^* l'insieme delle stringhe finite su questo alfabeto, cioè un insieme in naturale biezione con $\bigcup_{n\in\mathbb{N}} \Sigma^n$. Ricordiamo che è numerabile. Con λ denoteremo la stringa vuota.

Definizione 1.1. L'ordine **quasi-lessicografico** su Σ^* è un ordine totale che ordina le stringhe innanzitutto per lunghezza crescente e poi, in caso di pari lunghezza, in base all'ordine lessicografico.

È facile verificare che questo è un buon ordinamento. I primi elementi sono quindi λ, 0, 1, 00, 01, ... Viene naturale la seguente definizione.

Definizione 1.2. Chiamiamo $\mathrm{str} : \mathbb{N} \to \Sigma^*$ la corrispondenza biunivoca tra i naturali e Σ^* indotta dall'ordine quasi-lessicografico.

Cioè $\mathrm{str}(0) = \lambda$, $\mathrm{str}(1) = 0$, $\mathrm{str}(2) = 1$, $\mathrm{str}(3) = 00$, $\mathrm{str}(4) = 01$, ...

Osservazione. Chiamiamo $\mathrm{bin} : \mathbb{N}_+ \to \Sigma^*$ la funzione che ad ogni numero naturale positivo associa la sua rappresentazione in base 2 (senza zeri all'inizio). Per ogni $n \in \mathbb{N}$ vale

$$\mathrm{bin}(n+1) = 1\,\mathrm{str}(n) \tag{1.1}$$

Inoltre vale

$$|\,\mathrm{bin}(n)| = \lfloor \log(n) \rfloor + 1 \tag{1.2}$$

$$|\,\mathrm{str}(n)| = \lfloor \log(n+1) \rfloor \tag{1.3}$$

Fissiamo una codifica calcolabile delle coppie di stringhe, cioè una funzione $\langle \cdot, \cdot \rangle : \Sigma^* \times \Sigma^* \to \Sigma^*$ biettiva. Possiamo prendere, ad esempio, quella indotta dal dovetailing su $\mathbb{N}^2$ tramite la corrispondenza biunivoca str.

Calcolabilità

Tratteremo solamente funzioni parziali calcolabili binarie, da $\Sigma^* \times \Sigma^*$ a Σ^*. Fissiamo ora, una volta per tutte, una enumerazione accettabile delle funzioni parziali calcolabili binarie: ϕ_i con $i \in \mathbb{N}$. Vale quindi il teorema smn ed esiste un interprete ϕ_u per cui

$$\phi_u(\mathrm{str}(i), \langle x, y \rangle) = \phi_i(x, y) \tag{1.4}$$

Questa ϕ_u non è necessariamente unica.

Capiterà che ci prenderemo qualche leggera libertà nell'uso delle funzioni parziali calcolabili ma queste imprecisioni saranno solamente formali e sempre facilmente risolvibili. Ad esempio useremo str e bin (si osservi che

sono calcolabili) per definire altre funzioni parziali calcolabili anche se sono funzioni unarie e quindi non rientrano nel quadro appena descritto.

Sia $\varphi : \Sigma^* \times \Sigma^* \to \Sigma^*$ parziale calcolabile. Se la valutazione di φ su (x, y) diverge scriveremo che $\varphi(x, y) \uparrow$, mentre se converge scriveremo che $\varphi(x, y) \downarrow$. In tal caso (e solo in quello) sarà $(x, y) \in \operatorname{dom} \varphi$. Se scriviamo $\varphi(x, y) = z$ assumiamo implicitamente che $\varphi(x, y) \downarrow$.

Teoria dell'informazione classica

Definizione 1.3. Diremo che $A \subseteq \Sigma^*$ è **senza prefissi** se per ogni $x, y \in A$, $x \neq y$, si ha che x non è un prefisso di y.

Teorema 1.1 (Kraft)**.** *Sia $A = \{x_1, \ldots, x_m\} \subseteq \Sigma^*$ un insieme finito senza prefissi. Posto $n_k = |x_k|$ si ha*

$$\sum_{k=1}^{m} 2^{-n_k} \leq 1 \tag{1.5}$$

Viceversa, dati $n_1, \ldots, n_m$ che soddisfano (1.5) *esiste $A = \{x_1, \ldots, x_m\} \subseteq \Sigma^*$ senza prefissi con $|x_k| = n_k$.*

Definizione 1.4. Sia $A : X \to Y$ una variabile aleatoria, con Y al più numerabile, e sia P la misura di probabilità di X. Denotiamo con P_A la misura di probabilità indotta da A su Y, cioè $P_A(S) = P(A^{-1}(S))$. Allora l'**entropia di Shannon** di A è

$$\mathcal{H}(A) = -\sum_{y \in Y} P_A(y) \log P_A(y) \tag{1.6}$$

Capitolo 2

Complessità algoritmica

In questo capitolo ci proponiamo di introdurre e analizzare una misura di complessità per le stringhe: la lunghezza della più corta descrizione della stringa in un fissato formalismo. Ad esempio, in linguaggio naturale, 00000000 può essere descritto da "8 volte 0", 10101001 può essere descritto da "13^2 in base 2" e 00101010 può essere descritto da "il codice ASCII dell'asterisco". Queste non sono necessariamente le descrizioni minime.

Al fine di ottenere la massima generalità non ci limiteremo ad uno specifico linguaggio ma considereremo tutti i formalismi con semantica univoca e calcolabile, cioè per i quali le espressioni valide possono essere effettivamente valutate. Tutti i linguaggi di programmazione appartengono a questa famiglia. Il linguaggio naturale purtroppo ne è escluso, per via della sua inerente ambiguità (si consideri il paradosso di Berry: "la più piccola stringa non descrivibile con meno di dodici parole" è una descrizione con 11 parole).

Lavoreremo dunque con le funzioni parziali calcolabili, da $\Sigma^* \times \Sigma^*$ a Σ^*. Se φ è una tale funzione, interpreteremo l'espressione $\varphi(x, y)$ come il calcolo effettuato dalla *macchina* φ su cui è in esecuzione il *programma* x istanziato sull'*input* y. In termini più tecnologici, possiamo considerare φ come l'*hardware*, x come il *software* e y come l'*input*.

Ignoriamo dunque il modello di calcolo rispetto a cui abbiamo definito la calcolabilità (macchine di Turing, macchine RAM, funzioni ricorsive, lambda

calcolo, ...) e il modo in cui abbiamo ottenuto l'enumerazione accettabile effettiva (enumerazione di Gödel, codifica di Turing, linguaggi di programmazione, ...), qualsiasi essi siano, e consideriamo le funzioni parziali calcolabili come semplici enti matematici. Ciò che intendiamo dire, con la distinzione in hardware e software, è che ciascuna funzione parziale calcolabile definisce un suo proprio modello di calcolo, eventualmente non Turing-completo, che automaticamente determina un linguaggio e definisce una semantica in base alle coppie (x, y) che appartengono al dominio della funzione.

Segue un esempio che speriamo chiarisca le idee. Esisterà una φ parziale calcolabile che simulerà il comportamento di una calcolatrice tascabile. Avrà come linguaggio le espressioni aritmetiche sugli interi e sull'indeterminata x, opportunamente codificate sull'alfabeto binario; avrà come semantica la valutazione dell'espressione data come primo argomento (cioè come software) in cui la x è stata istanziata con il numero naturale codificato dal secondo argomento (cioè dall'input). È ovviamente un modello molto semplice e poco potente, non Turing-completo.

Ci siamo soffermati a spiegare questa interpretazione perché è un approccio un po' bizzarro, ma ci torna molto comodo. In particolare ci permetterà di trattare in modo estremamente generale tutti i possibili modelli di calcolo (e quindi tutti i possibili linguaggi effettivamente valutabili) e, considerando il codice sorgente come descrizione, è facile misurarne la dimensione prendendo la lunghezza della stringa passata come primo argomento.

Un approccio alternativo sarebbe potuto essere prendere le funzioni parziali calcolabili unarie (invece che binarie), enumerarle con delle stringhe (invece che con degli interi) e considerare la lunghezza di queste stringhe per la complessità. Vista l'arbitrarietà dell'enumerazione i risultati sarebbero valsi per qualsiasi enumerazione accettabile e quindi per qualsiasi modello di calcolo Turing-completo. Col nostro approccio, però, possiamo trattare modelli non completi (e.g. "hardware specializzato") senza alcuna difficoltà. Ci permetterà inoltre di osservare subito che tutti i modelli di calcolo universali sono sostanzialmente equivalenti per i nostri fini.

2.1 Complessità di Kolmogorov

Vediamo ora una prima definizione del tipo di misura di complessità che stiamo cercando. È la prima anche in ordine storico, introdotta in [5]. Vi premettiamo, però, il concetto di funzione universale.

Definizione 2.1. Una $\psi : \Sigma^* \to \Sigma^*$ parziale calcolabile si dice **universale** se per ogni $\varphi : \Sigma^* \to \Sigma^*$ parziale calcolabile esiste una costante $c \geq 0$ naturale tale che: per ogni $x, y \in \Sigma^*$ con $\varphi(x, y) \downarrow$ esiste $\tilde{x} \in \Sigma^*$ tale che $\psi(\tilde{x}, y) = \varphi(x, y)$ e $|\tilde{x}| \leq |x| + c$.

Teorema 2.1. *Esiste una $\psi : \Sigma^* \times \Sigma^* \to \Sigma^*$ parziale calcolabile universale.*

Utilizzando l'interprete ϕ_u è molto facile dimostrarlo costruttivamente.

Traccia di dimostrazione. Possiamo costruire una $\psi : \Sigma^* \times \Sigma^* \to \Sigma^*$ nel seguente modo

$$\psi(0^i 1 x, y) = \phi_u(\mathrm{str}(i), \langle x, y \rangle) = \phi_i(x, y)$$

Sia φ parziale calcolabile, allora sarà $\varphi = \phi_i$ per un certo $i \in \mathbb{N}$, perciò $\tilde{x} = 0^i 1 x$ e quindi $|\tilde{x}| = |x| + i + 1$. □

Nel seguito denoteremo con ψ una funzione universale fissata arbitrariamente. Poiché non assumeremo o imporremo mai struttura particolare su tale funzione i risultati enunciati varranno per qualsiasi funzione universale, e quindi per tutte esse.

Possiamo ora procedere a definire la misura di complessità di Kolmogorov.

Definizione 2.2. Sia $\varphi : \Sigma^* \times \Sigma^* \to \Sigma^*$ parziale calcolabile.

La **complessità di Kolmogorov** associata a φ è la funzione parziale $K_\varphi : \Sigma^* \to \mathbb{N}$

$$K_\varphi(x) = \min\{|u| \mid u \in \Sigma^*, \varphi(u, \lambda) = x\} \tag{2.1}$$

La **complessità condizionale di Kolmogorov** associata a φ è la funzione parziale $K_\varphi : \Sigma^* \times \Sigma^* \to \mathbb{N}$

$$K_\varphi(x/y) = \min\{|u| \mid u \in \Sigma^*, \varphi(u, y) = x\} \tag{2.2}$$

Inoltre poniamo $K(x) = K_\psi(x)$ e $K(x/y) = K_\psi(x/y)$.

Osservazione. Nonostante K_φ in generale possa non essere definita su certi input, K è una funzione totale poiché ψ, che è universale, è in grado di simulare tutte le funzioni costanti.

Se una certa K_φ non è definita su un valore x, converremo che $K_\varphi(x) = \infty$. Faremo lo stesso per la complessità condizionale.

Vale $K_\varphi(x) = K_\varphi(x/\lambda)$.

Vediamo ora il motivo per cui abbiamo preferito rimanere estremamente generali nella nostra impostazione di modello di calcolo: ciò, oltre a non introdurre differenze (asintoticamente) significative tra i vari modelli, ci offre qualche intuizione in più sulle relazioni tra funzioni universali.

Teorema 2.2 (invarianza)**.** *Per ogni $\varphi : \Sigma^* \times \Sigma^* \to \Sigma^*$ parziale calcolabile vale*

$$K(x) \leq K_\varphi(x) + O(1) \tag{2.3}$$

$$K(x/v) \leq K_\varphi(x/v) + O(1) \tag{2.4}$$

Dimostrazione. È una conseguenza immediata della definizione di funzione universale. □

Corollario 2.3. *Sia $\omega : \Sigma^* \times \Sigma^* \to \Sigma^*$ un'altra funzione parziale calcolabile universale. Allora*

$$|K(x) - K_\omega(x)| \leq O(1) \tag{2.5}$$

$$|K(x/y) - K_\omega(x/y)| \leq O(1) \tag{2.6}$$

Dimostrazione. Basta applicare il teorema precedente prima in un senso e poi nell'altro. □

Definizione 2.3. La **complessità congiunta** di $x, y \in \Sigma^*$ rispetto a φ è

$$K_\varphi(x, y) = K_\varphi(\langle x, y \rangle) \tag{2.7}$$

Passiamo ora a discutere di una formula molto importante: la regola della catena. Ci dice che determinare interamente un oggetto non è più complesso che determinarlo un pezzo alla volta, avendo a disposizione a ogni passo le parti già calcolate.

Teorema 2.4 (regola della catena)**.** *Vale*

$$K(x, y) \leq K(x) + K(y/x) + O(\log K(x, y)) \tag{2.8}$$

L'idea della dimostrazione è di determinare una funzione che costruisce (x, y) se le vengono date le istruzioni per costruire, indipendentemente, il valore x e il valore y conoscendo x. Queste istruzioni vanno unite in un'unica stringa ma, purtroppo, non è possibile semplicemente giustapporle poiché poi la funzione non saprebbe come separarle correttamente. Vanno codificate in qualche modo, e ciò è possibile solo al costo di un incremento logaritmico della lunghezza totale.

Introdurremo un nuovo sistema di codifica delle coppie di stringhe, pensato ad-hoc per dimostrare elegantemente l'efficienza richiesta per questo risultato.

Dimostrazione. Introduciamo un codice istantaneo per i numeri naturali, cioè una funzione $\pi : \mathbb{N} \to \Sigma^*$ iniettiva tale che $\pi(\mathbb{N})$ sia senza prefissi. Useremo il codice che a $n \in \mathbb{N}$ con $\operatorname{str}(n) = x_1 \cdots x_k$ $(x_i \in \Sigma)$ associa $\pi(n) = 0x_1 \cdots 0x_k 1$. È facile osservare che è istantaneo, e quindi è anche univocamente decodificabile. La decodifica procede così: legge un carattere dell'input, se è 0 legge anche il successivo, lo aggiunge in coda al risultato e ripete, se è 1 termina e restituisce str^{-1} del risultato ottenuto. Notiamo che $|\pi(n)| = 2|\operatorname{str}(n)| + 1$. Usando questa π definiamo ora una $f : \Sigma^* \times \Sigma^* \to \Sigma^*$ iniettiva tale che $f(u, v) = \pi(|u|)uv$. L'inversa (che sarà una funzione parziale) non fa che ricavare $\pi(|u|)$ (è possibile senza ambiguità, in quanto π è codice istantaneo) grazie a cui potrà determinare quanti dei caratteri rimanenti appartengono alla prima stringa (i primi $|u|$) e quanti alla seconda (i restanti). Vale

$$|f(u, v)| = |\pi(|u|)| + |u| + |v| = |u| + |v| + 2|\operatorname{str}(|u|)| + 1$$

Con questi strumenti definiamo ora $\varphi(a, b)$ nel seguente modo: assumendo $a = f(u, v)$ invertiamo f e ricaviamo u e v (divergendo se non è possibile, ad esempio perché a non comincia con un elemento di $\pi(\mathbb{N})$), poi calcoliamo $x = \psi(u, \lambda)$ e $y = \psi(v, x)$ e restituiamo $\langle x, y\rangle$.

Sia u la stringa su cui si ottiene il minimo della definizione di $K(x)$ (cioè $\psi(u, \lambda) = x$ e $|u| = K(x)$). Sia v lo stesso per $K(y/x)$ (cioè $\psi(v, x) = y$ e $|v| = K(x/y)$). Allora $\varphi(f(u, v), \lambda) = \langle x, y\rangle$ e quindi

$$\begin{aligned} K(x, y) &\leq K_\varphi(x, y) + c \\ &\leq |f(u, v)| + c \\ &= |u| + |v| + 2|\operatorname{str}(|u|)| + c + 1 \\ &\leq K(x) + K(x/y) + 2\log(K(x) + 1) + c + 1 \end{aligned}$$ □

Nella regola della catena vale, in realtà, l'uguaglianza (a meno del termine logaritmico). Questo risultato si trova come teorema 5.2 in [9].

Ricaviamo ora qualche altra utile formula.

Proposizione 2.5. *Sia $f : \Sigma^* \to \Sigma^*$ una funzione calcolabile. Valgono*

$$K(x) \leq |x| + O(1) \tag{2.9}$$

$$K(x/x) = O(1) \tag{2.10}$$

$$K(x/y) \leq K(x) + O(1) \tag{2.11}$$

$$K(x, y) \leq K(x) + K(y) + O(\log K(x, y)) \tag{2.12}$$

$$K(f(x)) \leq K(x) + O(1) \tag{2.13}$$

$$K(y, x) \leq K(x, y) + O(1) \tag{2.14}$$

$$K(x) \leq K(x, y) + O(1) \tag{2.15}$$

La prima formalizza l'idea che si può scrivere un programma che contiene la codifica incompressa esplicita di una stringa e aggiungere un costrutto (di lunghezza costante) per "stampare" la stringa. Questo programma sarà un maggiorante per la complessità di Kolmogorov.

La seconda e la terza risultano intuitivamente molto ovvie: la complessità richiesta per determinare un oggetto se esso è già dato è costante; la

complessità non può aumentare se si mettono a disposizione informazioni aggiuntive (anche se esse sono del tutto irrilevanti). La quarta è una sorta di subadditività a meno di un termine logaritmico.

La quinta si può interpretare come l'intuizione che applicando una funzione calcolabile l'informazione non aumenta: al più diminuisce e, con essa, diminuisce la complessità della stringa. La sesta dice che l'ordine degli elementi in una coppia è irrilevante e la settima dice che la complessità richiesta per determinare una parte di un oggetto è minore di quella richiesta per determinarlo tutto.

Dimostrazione. Per (2.9) consideriamo $\varphi(x, y) = x$, per cui $K_\varphi(x) = |x|$, e applichiamo il teorema di invarianza 2.2.

Per (2.10) consideriamo $\varphi(x, y) = y$, per cui $K_\varphi(x/x) = 0$, e applichiamo il teorema di invarianza 2.2.

Per (2.11) consideriamo $\varphi(x, y) = \psi(x, \lambda)$ (quindi φ ignora il suo input), per cui $K_\varphi(x/y) = K(x)$, e applichiamo il teorema di invarianza 2.2.

Per (2.12) basta combinare (2.8) e (2.11).

Per (2.13) consideriamo $\varphi(x, y) = f(\psi(x, y))$ e, per il teorema di invarianza 2.2, vale

$$K(f(x)) \leq K_\varphi(f(x)) + O(1) \leq K(x) + O(1)$$

Prendendo $f(\langle x, y\rangle) = \langle y, x\rangle$, (2.14) è un'applicazione di (2.13).

Prendendo $f(\langle x, y\rangle) = x$, (2.15) è un'applicazione di (2.13). □

Definizione 2.4. La **mutua informazione algoritmica** di $x, y \in \Sigma^*$ rispetto a φ è

$$I_\varphi(x : y) = K_\varphi(y) - K_\varphi(y/x) \tag{2.16}$$

Poniamo $I(x : y) = I_\psi(x : y)$.

Proposizione 2.6. *Valgono*

$$I(x : y) + O(1) \geq 0 \tag{2.17}$$

$$I(x : y) \leq K(x) + K(y) - K(x, y) + O(\log K(x, y)) \tag{2.18}$$

$$I(x : x) = K(x) + O(1) \tag{2.19}$$

$$I(x : \lambda) = O(1) \tag{2.20}$$

$$I(\lambda : x) = O(1) \tag{2.21}$$

Dimostrazione. Per (2.17) applichiamo (2.11) alla definizione.

Per (2.18) applichiamo (2.8) alla definizione.

Per (2.19) applichiamo (2.10) alla definizione.

Per (2.20) basta ricordare $K(x/\lambda) = K(x)$.

Per (2.21) basta prendere la $\varphi(x, y) = \lambda$ e applicare il teorema di invarianza 2.2. □

2.2 Complessità di Chaitin

La direzione in cui si è mossa la ricerca moderna negli ultimi decenni è stata la complessità di Chaitin, indubbiamente ispirata a quella di Kolmogorov come concezione ma con un dettaglio "tecnico" differente che le garantisce interessanti proprietà aggiuntive. L'osservazione da cui Chaitin è partito è stata che tutti i linguaggi di programmazione hanno bisogno di un modo per identificare la terminazione del codice sorgente di un programma o, equivalentemente, la sua lunghezza. Ciò è stato implementato in vari modi: costrutti di apertura e chiusura di blocchi di codice (i.e. il codice è terminato alla chiusura del blocco annidato più esterno), indicatore esplicito di terminazione (che però è una stringa di caratteri dell'alfabeto) oppure dichiarazione esplicita della lunghezza (all'interno dello stesso codice).

Queste osservazioni hanno portato alla ridefinizione del concetto di complessità, che è stato rafforzato da una condizione aggiuntiva: l'insieme dei programmi che vengono considerati (cioè quelli delle cui lunghezze si cerca il

minimo) non può contenere due stringhe che sono l'una un prefisso dell'altra. Formalmente lo si esprime nel seguente modo.

Definizione 2.5. Sia $C : \Sigma^* \times \Sigma^* \to \Sigma^*$ parziale calcolabile. Chiameremo **sezione di** C **lungo** y, con $y \in \Sigma^*$, la funzione parziale calcolabile $C_y : \Sigma^* \to \Sigma^*$ per cui $C_y(x) = C(x, y)$.

Definizione 2.6. Sia $C : \Sigma^* \times \Sigma^* \to \Sigma^*$ parziale calcolabile. Diremo che C è **di Chaitin** se $\operatorname{dom} C_y$ è senza prefissi per ogni $y \in \Sigma^*$.

La richiesta dell'auto-delimitazione dei codici sorgente rispetto ad un input fissato, come abbiamo appena fatto, è più debole della richiesta dell'auto-delimitazione dei codici sorgente rispetto a qualsiasi input (cioè imporre che $\bigcup_{y\in\Sigma^*} \operatorname{dom} C_y$ sia senza prefissi) ma sarà sufficiente per l'uso che ne faremo.

In maniera del tutto analoga al teorema 2.1 si dimostra il seguente teorema.

Teorema 2.7. *Esiste una* $U : \Sigma^* \times \Sigma^* \to \Sigma^*$ *parziale calcolabile universale di Chaitin.*

Nel seguito con U denoteremo una funzione universale di Chaitin fissata ad arbitrio. Anche in questo caso, vista l'arbitrarietà, i risultati che seguiranno varranno per tutte le funzioni universali di Chaitin.

Usando questa definizione, possiamo introdurre una nuova misura di complessità.

Definizione 2.7. Sia $C : \Sigma^* \times \Sigma^* \to \Sigma^*$ parziale calcolabile di Chaitin.

Il **programma minimale** rispetto a U di $x \in \Sigma^*$ è

$$x^* = \min\{u \in \Sigma^* \mid U(u, \lambda) = x\} \tag{2.22}$$

dove il minimo è preso rispetto all'ordine quasi-lessicografico.

La **complessità di Chaitin** associata a C è la funzione parziale $H_C : \Sigma^* \to \mathbb{N}$

$$H_C(x) = \min\{|u| \mid u \in \Sigma^*, C(u, \lambda) = x\} \tag{2.23}$$

La **complessità condizionale di Chaitin** associata a C è la funzione parziale $H_C : \Sigma^* \times \Sigma^* \to \mathbb{N}$

$$H_C(x/y) = \min\{|u| \mid u \in \Sigma^*, C(u, y^*) = x\} \tag{2.24}$$

Inoltre poniamo $H(x) = H_U(x)$ e $H(x/y) = H_U(x/y)$.

Vale $H(x) = |x^*|$. Nella definizione della complessità condizionale abbiamo passato y^* come secondo parametro invece di y per ottenere poi un risultato più elegante nella regola della catena. Questo dettaglio non si rivelerà tuttavia cruciale a tale scopo in quanto anche se avessimo proceduto in maniera simile per la complessità di Kolmogorov (utilizzando i programmi minimali rispetto a ψ) non sarebbe cambiata significativamente la formulazione della regola della catena.

Anche per la complessità di Chaitin si ha un teorema di invarianza.

Teorema 2.8 (invarianza)**.** *Per ogni $C : \Sigma^* \times \Sigma^* \to \Sigma^*$ parziale calcolabile di Chaitin vale*

$$H(x) \leq H_C(x) + O(1) \tag{2.25}$$

$$H(x/y) \leq H_C(x/y) + O(1) \tag{2.26}$$

In maniera naturale si ridefinisce anche la complessità congiunta di Chaitin. Vediamo però subito la prima sostanziale differenza tra le due definizioni di complessità: la regola della catena, in cui il termine logaritmico in questo caso si riduce ad una costante.

Teorema 2.9. *Vale*

$$H(x, y) = H(x) + H(y/x) + O(1) \tag{2.27}$$

Mostriamo innanzitutto una direzione, la stessa che abbiamo dimostrato per la complessità di Kolmogorov. Per dimostrare la disuguaglianza inversa occorreranno strumenti più sofisticati, che introdurremo subito dopo.

Premettiamo due osservazioni.

Osservazione. In quanto U è parziale calcolabile il suo dominio è ricorsivamente enumerabile e quindi così è anche U_λ.

Osservazione. Come K, anche H è una funzione totale (sia la versione "standard" che quella condizionale). Inoltre U_y è suriettiva per ogni $y \in \Sigma^*$.

Dimostrazione del $\leq$. L'idea è simile a quella usata per il teorema 2.4. Vedremo subito come entra in gioco l'auto-delimitazione dei programmi per ridurre il termine logaritmico a un termine costante.

Sia $C(u, v)$ la funzione che opera nel seguente modo: inizia a generare una ad una tutte le stringhe di U_λ e si arresta quando ne trova una, chiamiamola a, che è prefisso di u, cioè $u = ab$, con $b \in \Sigma^*$; allora calcola e restituisce $\langle U(a, \lambda), U(b, a)\rangle$.

Sia $r = x^*$ (quindi $|r| = H(x)$ e $U(r, \lambda) = x$) e sia $s \in \Sigma^*$ su cui si ottiene il minimo della definizione di $H(y/x)$ (cioè $|s| = H(y/x)$ e $U(s, r) = y$). Tali r e s esistono per la precedente osservazione. Vediamo ora che $C(rs, \lambda) = \langle x, y\rangle$ in quanto la a della definizione sarà proprio r (poiché U è di Chaitin e quindi il dominio di U_λ è senza prefissi) e quindi b sarà s.

Vale quindi

$$\begin{aligned} H(x, y) &\leq H_C(x, y) + O(1) \\ &\leq |rs| + O(1) \\ &= |r| + |s| + O(1) \\ &= H(x) + H(y/x) + O(1) \end{aligned}$$ □

2.2.1 Probabilità d'arresto

Sulle funzioni di Chaitin possiamo effettuare un'importante costruzione probabilistica che ci aiuterà a dimostrare fondamentali proprietà. L'idea è di quantificare la probabilità che una certa funzione di Chaitin termini e produca un certo output. Per poterla definire formalmente dobbiamo dare una struttura di spazio di probabilità all'insieme delle stringhe di input per poi vedere la funzione di Chaitin come una variabile aleatoria da questo spazio all'insieme delle stringhe di output.

In termini intuitivi, faremo corrispondere a ciascuna stringa di Σ^* un sotto-intervallo di $[0,1)$, e la probabilità della stringa sarà la misura di Lebesgue di quell'intervallo. L'intervallo associato alla stringa $u \in \Sigma^*$ sarà l'insieme dei numeri in $[0,1)$ la cui parte frazionaria dello sviluppo in base 2 inizia con u, cioè ha u come prefisso. Quindi alla stringa 0 sarà associato l'intervallo $[0, 1/2)$, alla stringa 10 l'intervallo $[1/2, 3/4)$, mentre a 0110 l'intervallo $[3/8, 7/16)$. Usiamo intervalli semiaperti perché ci limitiamo alle rappresentazioni frazionarie proprie e scegliamo di ignorare quelle improprie. Tuttavia ciò è irrilevante, poiché la differenza è un insieme con un solo elemento e quindi di misura nulla. Notiamo che negli esempi appena fatti la probabilità associata ad una stringa $u \in \Sigma^*$ è $2^{-|u|}$. Sarà effettivamente così, ma ci arriveremo attraverso un'altra costruzione, più generale.

Consideriamo l'insieme Σ^ω delle stringhe infinite di alfabeto Σ, che è come dire le successioni in Σ. Su di esso definiamo una base numerabile di aperti, uno per ogni $u \in \Sigma^*$. Infine prenderemo la σ-algebra di Borel costruita su questi aperti, e la misura di probabilità associata, come spazio di probabilità. L'aperto associato a $u \in \Sigma^*$, che denoteremo con $u\Sigma^\omega$, è l'insieme di tutte le stringhe infinite che ammettono u come prefisso e ha misura di probabilità $2^{-|u|}$.

Una prima proprietà elementare è $\lambda\Sigma^\omega = \Sigma^\omega$ e quindi entrambi hanno probabilità 1. Inoltre:

$$u \text{ prefisso di } v \iff u\Sigma^\omega \supseteq v\Sigma^\omega$$
$$u \text{ e } v \text{ non sono prefissi l'uno dell'altro} \iff u\Sigma^\omega \cap v\Sigma^\omega = \emptyset$$

e queste proprietà si sposano perfettamente con le caratteristiche delle funzioni di Chaitin, cioè di avere un dominio senza prefissi. Infatti significa che gli aperti corrispondenti al dominio di una sezione di una funzione di Chaitin sono a due a due disgiunti, e quindi la somma delle loro probabilità sarà uguale alla probabilità della loro unione, che sarà minore o uguale a 1.

Formalizziamolo introducendo una opportuna variabile aleatoria. Fissiamo $C : \Sigma^* \times \Sigma^* \to \Sigma^*$ di Chaitin e definiamo $X_C : \Sigma^\omega \to \Sigma^*$ parziale che

su $u \in \Sigma^\omega$ opera nel seguente modo: se esiste $v \in \Sigma^*$ prefisso di u per cui $C(v, \lambda) \downarrow$ (vale a dire che v appartiene al dominio della sezione di C lungo λ) allora restituisce $C(v, \lambda)$, altrimenti diverge. Poiché C è di Chaitin tale v, se esiste, è unico e quindi X_C è ben definita. È una variabile aleatoria poiché la preimmagine di $x \in \Sigma^*$ è l'unione di una famiglia al più numerabile di aperti elementari $u\Sigma^\omega$. Come dicevamo sono disgiunti e quindi possiamo finalmente definire:

$$P_C(x) = P(X_C^{-1}(x)) = P\left(\bigcup_{z \in C_\lambda^{-1}(x)} z\Sigma^\omega\right) = \sum_{z \in C_\lambda^{-1}(x)} P(z\Sigma^\omega) = \sum_{z \in C_\lambda^{-1}(x)} 2^{-|z|} \quad (2.28)$$

In maniera perfettamente analoga definiamo

$$P_C(x/y) = \sum_{z \in C_{y^*}^{-1}(x)} 2^{-|z|} \quad (2.29)$$

Queste funzioni rappresentano, nel senso che abbiano appena descritto, la probabilità che, scelta in maniera uniforme la stringa che rappresenta il codice del programma, l'esecuzione di C su quel programma (con input λ o y, a seconda dei casi) termini e restituisca un certo output. I due valori vengono chiamati **probabilità algoritmica** e **probabilità algoritmica condizionale**.

Dopo aver faticosamente introdotto e giustificato questi concetti vediamone alcune importanti applicazioni. Le dimostrazioni e i risultati intermedi minori sono troppo laboriosi per essere riportati nella loro interezza. Per una trattazione appropriata rimandiamo a [1].

Il primo risultato è una riformulazione della disuguaglianza di Kraft e del suo teorema sui codici istantanei.

Teorema 2.10 (Kraft–Chaitin)**.** *Sia $f : \mathbb{N} \to \Sigma^* \times \mathbb{N}$ parziale calcolabile il cui dominio è della forma $\{1, \ldots, m\}$. Per ogni $k \in \operatorname{dom} f$ poniamo $(x_k, n_k) = f(k)$. Se vale*

$$\sum_{k=1}^{m} 2^{-n_k} \leq 1 \quad (2.30)$$

allora possiamo costruire una $C : \Sigma^* \times \Sigma^* \to \Sigma^*$ *di Chaitin tale che per ogni* $k \in \operatorname{dom} f$ *esiste una* u_k *di lunghezza* n_k *tale che* $C(u_k, \lambda) = x_k$.

Da esso si può derivare un altro teorema. Fornisce la connessione tra complessità di Chaitin e probabilità algoritmica, ed è quindi di cruciale importanza.

Teorema 2.11. *Valgono*

$$H(x) = -\log P(x) + O(1) \tag{2.31}$$

$$H(x/y) = -\log P(x/y) + O(1) \tag{2.32}$$

Vediamone una applicazione: la regola della catena per la complessità di Chaitin, nella direzione che ancora ci resta da dimostrare. Va premesso un lemma.

Lemma 2.12. *Esiste una* $C : \Sigma^* \times \Sigma^* \to \Sigma^*$ *di Chaitin e una costante* $c > 0$ *tale che*

$$H_C(y/x) = H(x, y) - H(x) + c \tag{2.33}$$

Dimostrazione del $\geq$ *del teorema 2.9.* Dal lemma 2.12 otteniamo una $C : \Sigma^* \times \Sigma^* \to \Sigma^*$ per cui vale (2.33). Ne viene, per il teorema di invarianza 2.8

$$H(x, y) = H_C(y/x) + H(x) - c \geq H(y/x) + H(x) + O(1) \qquad \square$$

Notiamo che, poiché per la complessità di Chaitin la regola della catena ha un termine costante invece che logaritmico, riusciamo a "tradurre simbolicamente" tutte le formule notevoli dell'entropia di Shannon in termini di complessità di Chaitin, sempre a meno di un termine costante. Cioè: tra i concetti corrispondenti delle due teorie valgono le stesse relazioni. Questo è un risultato strabiliante, poiché le due quantità misurano concetti intuitivamente molto diversi: una misura l'incertezza sull'esito della "estrazione" di un valore da uno spazio di probabilità, l'altra misura la complessità intrinsecamente necessaria per rappresentare algoritmicamente una stringa. In altri termini possiamo dire che l'entropia di Shannon è la minima descrizione (in

media) necessaria ad identificare un oggetto conoscendo solo lo spazio di probabilità da cui è estratto ed ignorandone la struttura; mentre la complessità di Kolmogorov è, viceversa, la minima descrizione necessaria ad identificare un oggetto considerando esclusivamente la sua struttura e tralasciando la sua origine. Tuttavia il collegamento cruciale tra le due ce lo dà proprio la costruzione probabilistica effettuata sulla complessità di Chaitin, e in particolare il teorema 2.11, da cui si può dedurre che l'entropia di Shannon di una variabile aleatoria è uguale al valore medio della complessità di Kolmogorov dei valori che può assumere. Una volta formulato tale risultato è facile intuire perché vale: data una variabile aleatoria possiamo costruire una funzione parziale calcolabile che produce gli stessi valori e che richiede programmi corti per valori molto probabili e programmi lunghi per valori poco probabili, e poi applicare il teorema di invarianza. Per un confronto più sistematico tra le complessità di Kolmogorov e di Chaitin e l'entropia di Shannon si veda [4].

Capitolo 3

Incomprimibilità

Abbiamo introdotto lo strumento della complessità algoritmica al fine di tentare di catturare formalmente l'intuizione di casualità di una stringa.

Anche già dare una definizione in lingua naturale di tale concetto non è facile: nel cercare di focalizzarlo ci accorgiamo che ci sono vari aspetti, di differente natura, che vorremmo che tali oggetti possedessero. Se li vediamo come "registrazioni" dei risultati di estrazioni da un processo casuale (cioè esiti di una sequenza di variabili aleatorie i.i.d.), ci aspettiamo che valgano tutte le proprietà della teoria della probabilità. Ad esempio, per soddisfare la legge dei grandi numeri, il loro numero di zeri deve essere *circa* uguale al loro numero di uni. Già tale considerazione, nello specifico l'uso del "circa", porta con sé due fondamentali conseguenze. In primo luogo, il concetto di casualità ha poco senso per stringhe corte (come caso limite: le stringhe 0 e 1 sono casuali?): inizierà ad assumere significato per stringhe sufficientemente lunghe. In secondo luogo, è meglio chiedersi *quanto* un oggetto sia casuale, che chiedersi *se* è casuale: rendere quindi la casualità una misura invece di una proprietà.

Un'altra caratterizzazione delle stringhe casuali può essere l'assenza di regolarità. Una stringa molto regolare è poco casuale, e viceversa. La complessità di Kolmogorov è un possibile modo per misurare la regolarità di una stringa poiché sfruttando la regolarità si può trovare un programma per pro-

durre la stringa il cui codice sorgente è più corto della stringa stessa. Ne viene che una proprietà (meno intuitiva) delle stringhe casuali deve essere l'**incomprimibilità algoritmica**.

Definizione 3.1. Sia $t \in \mathbb{N}$. Diremo che $x \in \Sigma^*$ è **t-incomprimibile**, e scriveremo $x \in \mathrm{NC}_t$, se

$$K(x/|x|) \geq |x| - t \tag{3.1}$$

Possiamo identificare due "informazioni" in una stringa: la sua lunghezza e l'irregolarità del suo pattern di zeri e uni. Poiché noi siamo interessati esclusivamente nella seconda, nella definizione abbiamo utilizzato la complessità condizionale rispetto alla lunghezza, per cercare di eliminarne gli effetti sulla misura.

Si osserva subito che la complessità condizionale induce una definizione più forte rispetto alla complessità incondizionale: se $x \in \mathrm{NC}_t$ allora, poiché esiste $c \geq 0$ intero tale che $K(x/|x|) \leq K(x) + c$, vale $K(x) \geq |x| - t - c$. Questa costante diventa trascurabile quando si considerano stringhe lunghe.

Abbiamo leggermente abusato la definizione della complessità condizionale, permettendo al secondo argomento (l'input della computazione) di essere naturale invece di stringa. Possiamo permettercelo poiché questo parametro viene semplicemente passato alla funzione parziale calcolabile sottostante e non concorre in altro modo nella determinazione della complessità. Per risolvere formalmente questo problema basterebbe codificare questo naturale con str e wrappare la φ in una $\tilde{\varphi}$ tale che $\tilde{\varphi}(x, y) = \varphi(x, \mathrm{str}^{-1}(y))$.

Poiché ci sono poche stringhe corte (nel senso che il numero di stringhe cresce esponenzialmente con la loro lunghezza) ci saranno anche poche stringhe fortemente comprimibili. Quindi la maggior parte delle stringhe sarà incomprimibile. Questo rispecchia ciò che ci aspettiamo: una stringa scelta casualmente sarà casuale con altissima probabilità.

Poiché $\sum_{k=0}^{n-1} 2^k = 2^n - 1$ deve esistere almeno una stringa x di lunghezza n per cui $K(x/|x|) \geq n = |x|$, e quindi NC_0 contiene almeno una stringa di ogni lunghezza e di conseguenza lo stesso vale per ogni NC_t.

Similmente NC_1 contiene almeno $2^n - 2^{n-1} + 1$ stringhe di lunghezza n, per ogni $n \geq 1$. In generale NC_t contiene almeno $2^n - 2^{n-t} + 1$ stringhe di lunghezza n, per $n \geq t$. Un'ulteriore generalizzazione è data dal seguente teorema.

Teorema 3.1. *Sia $c > 0$ intero, $y \in \Sigma^*$ fissato e $A \subseteq \Sigma^*$ un insieme finito di cardinalità m. Allora per almeno $m(1 - 2^{-c}) + 1$ elementi x di A vale $K(x/y) \geq \log m - c$.*

Dimostrazione. Gli elementi con complessità inferiore a $\log m - c$ possono essere al massimo $\sum_{k=0}^{\log m - c - 1} 2^k = 2^{\log m - c} - 1 = m2^{-c} - 1$

Dunque ci sono almeno $m - m2^{-c} + 1$ elementi di A con complessità maggiore o uguale a $\log m - c$. □

Come interessante applicazione di questo teorema vediamo che il termine logaritmico nella regola della catena per la complessità di Kolmogorov non può essere eliminato. Fissiamo $n > 0$ intero e considerando l'insieme A delle coppie $\langle x, y \rangle$ con $|x| + |y| = n$, che contiene $2^n(n+1)$ elementi. Per il teorema 3.1 (con $c = 1$ e $y = \lambda$) esiste almeno un elemento con $K(x, y) \geq n + \log n - 1$. Poiché per (2.9) vale $K(x) + K(y) \leq |x| + |y| + k = n + k$, in conclusione

$$K(x, y) \geq K(x) + K(y) + \log n - k - 1$$

(questo vale per ogni n, e k non dipende da n)

In [6] si ritrova questa stessa impostazione dell'incomprimibilità, con ulteriori risultati.

Capitolo 4

Stocasticità

Come accennato nel capitolo precedente, una stringa può essere vista come il risultato di un processo stocastico discreto finito, vale a dire una sequenza finita di estrazioni da uno spazio di probabilità discreto. Assumendo una distribuzione di probabilità su questo spazio ci si può chiedere quanto sia "*tipica*" una certa sequenza di esiti, cioè in quale misura si avvicini alle aspettative prodotte dalla teoria della probabilità sugli esiti di esperimenti indipendenti. Nel nostro caso viene naturale prendere $\Sigma = \{0, 1\}$ con la distribuzione uniforme come spazio di probabilità, cioè il lancio di una moneta. Intuitivamente ci aspetteremmo quindi che la stringa

$$u_1 = 0100010110100111$$

rappresenti un tipico esito di 16 lanci, mentre la stringa

$$u_2 = 0000000000000000$$

sia un risultato meno tipico.

Purtroppo la teoria classica della probabilità non ci permette di distinguere tra le due, poiché sullo spazio Σ^{16} dotato della probabilità prodotto entrambe hanno probabilità 2^{-16}. Cerchiamo quindi altri strumenti per catturare formalmente l'essenza della nostra intuizione di tipicità.

Consideriamo a titolo di esempio la legge dei grandi numeri. Poiché ogni nuova estrazione è indipendente dalle precedenti ci aspettiamo che il numero

delle occorrenze di 0 sia tanto più vicino al numero delle occorrenze di 1 quanto più la stringa sia lunga. Chiamiamo $N_i(x)$ il numero delle occorrenze di i in x, con $i \in \Sigma$ e $x \in \Sigma^*$. Allora diciamo che $x \in \Sigma^*$ ha criticità ε se

$$\left| \frac{N_i(x)}{|x|} - \frac{1}{2} \right| = \varepsilon$$

Siamo portati a dire che una criticità bassa significa una stringa più vicina all'esito tipico che ci si aspetta da un processo aleatorio, e quindi una stringa "più casuale".

Lo stesso ragionamento vale per le occorrenze delle sottostringhe di due caratteri, di tre caratteri e di ogni altra lunghezza ragionevole, nonché per tutte le altre proprietà dei processi stocastici.

L'obiettivo di questo capitolo è di mostrare che l'incomprimibilità soddisfa queste nozioni di casualità. Si potrebbe dimostrare questo risultato direttamente, caso per caso. È tuttavia preferibile introdurre uno strumento generale che permetta di dimostrarlo una volta per tutte le proprietà stocastiche.

4.1 Test di Martin-Löf

L'idea di Martin-Löf, introdotta in [7], fu di definire un test statistico generico come una famiglia numerabile di *livelli di criticità*, ciascuno dei quali contiene tutte le stringhe la cui criticità è sopra un certo livello di soglia. Nella nostra esposizione seguiremo l'approccio di Calude e Chiţescu [1, 2].

Definizione 4.1. Un insieme ricorsivamente enumerabile $A \subseteq \Sigma^* \times \mathbb{N}_+$ è detto **test di Martin-Löf** se, posto $V_m = \{x \in \Sigma^* \mid (x, m) \in V\}$, vale:

- $V_{m+1} \subseteq V_m$ per ogni $m \geq 1$;
- $\#(\Sigma^n \cap V_m) \leq 2^{n-m}$ per ogni $n \geq 0$ e $m \geq 1$.

La seconda proprietà significa che, limitatamente alle stringhe di lunghezza n, il primo livello le può contenere tutte, il secondo solo la metà, il terzo solo un quarto, e così via.

Ne viene che se $(x, m) \in V$ allora $|x| \geq m$.

Vediamo l'esempio in cui la criticità è funzione della differenza tra il numero delle occorrenze di zero e di uno. Sia x una stringa di lunghezza n e sia $k = |N_0(x) - N_1(x)|$. Allora $(x, m) \in V$ se e solo se $k < g(n, m)$, dove $g(n, m)$ è il più piccolo intero l tale $\sum_{i=0}^{l} \binom{n}{i} > 2^{n-m}$. L'artificiosità di questa costruzione dipende dalla dimensione limite che abbiamo scelto per i nostri livelli critici, cioè 2^{n-m}. Una scelta differente avrebbe reso più naturale questo esempio. Tuttavia tutte le scelte sono sostanzialmente equivalenti, e vedremo a breve che la nostra si presta molto bene a mostrare la generalità dell'approccio di Martin-Löf.

Definizione 4.2. Ad ogni test di Martin-Löf V associamo la funzione di **livello critico** $m_V : \Sigma^* \to \mathbb{N}$ così definita:

$$m_V(x) = \max\left(\{0\} \cup \{m \geq 1 \mid (x, m) \in V\}\right) \tag{4.1}$$

Vale a dire, $m_V(x)$ è l'ultimo livello di V che contiene la stringa x, oppure zero se nessun livello la contiene.

È ben definita e risulta $m_V(x) \leq |x|$.

Definizione 4.3. Una stringa x è **q-casuale** rispetto al test di Martin-Löf V se $|x| \geq q$ e $m_V(x) \leq q$.

4.1.1 Rappresentabilità

Vediamo ora meglio come costruire un opportuno test di Martin-Löf che permetta di rappresentare un qualsiasi test di stocasticità esprimibile tramite un algoritmo. Questa è la massima generalità a cui possiamo aspirare.

Proposizione 4.1. *Sia $\varphi : \Sigma^* \times \mathbb{N} \to \Sigma^*$ parziale calcolabile. Poniamo*

$$V(\varphi) = \{(x, m) \in \Sigma^* \times \mathbb{N}_+ \mid K_\varphi(x/|x|) < |x| - m\} \tag{4.2}$$

Allora $V(\varphi)$ è un test di Martin-Löf.

Dimostrazione. Usando il predicato di Kleene e il teorema di proiezione si verifica che $V(\varphi)$ è ricorsivamente enumerabile. È banale osservare che $V_{m+1} \subseteq V_m$. Vediamo invece che

$$\begin{aligned}\#(A^n \cap V(\varphi)_m) &= \#\{x \in \Sigma^n \mid (x,m) \in V(\varphi)\} \\ &= \#\{x \in \Sigma^n \mid K_\varphi(x/n) < n-m\} \\ &\leq 2^{n-m} - 1\end{aligned}$$

□

Definizione 4.4. Sia V un test di Martin-Löf. Diremo che V è **rappresentabile** se esiste $\varphi : \Sigma^* \times \mathbb{N} \to \Sigma^*$ parziale calcolabile tale che $V = V(\varphi)$.

Proposizione 4.2. *Valgono:*

$$m_{V(\varphi)} > 0 \implies m_{V(\varphi)}(x) = |x| - K_\varphi(x/|x|) - 1 \tag{4.3}$$

$$m_{V(\varphi)} = 0 \iff 0 \geq |x| - K_\varphi(x/|x|) - 1 \tag{4.4}$$

In ogni caso

$$m_{V(\varphi)}(x) \geq |x| - K_\varphi(x/|x|) - 1 \tag{4.5}$$

Dimostrazione. Se $m_{V(\varphi)} > 0$ allora

$$\begin{aligned}m_{V(\varphi)}(x) &= \max(\{m \geq 1 \mid (x,m) \in V(\varphi)\}) \\ &= \max(\{m \geq 1 \mid K_\varphi(x/|x|) < |x| - m\}) \\ &= |x| - K_\varphi(x/|x|) - 1\end{aligned}$$

Se $m_{V(\varphi)} = 0$ allora $(x,1) \notin V(\varphi)$ e quindi $K_\varphi(x/|x|) \geq |x| - 1$. □

Non tutti i test di Martin-Löf sono rappresentabili. Infatti, considerando $V = \{(00,1),(01,1)\}$, che è un test di Martin-Löf, e supponendo per assurdo che esista una certa $\varphi : \Sigma^* \times \mathbb{N}_+ \to \Sigma^*$ per cui $V = V(\varphi)$, sarebbe necessario $K_\varphi(00/2) = 0$ e quindi $\varphi(\lambda,2) = 00$, ma anche $K_\varphi(01/2) = 0$ e quindi $\varphi(\lambda,2) = 01$, assurdo.

Tuttavia riusciamo ad avvicinarci a questo risultato con il seguente importante teorema:

Teorema 4.3. *Sia V un test di Martin-Löf e sia $u \in \Sigma^*$ con $u \neq \lambda$. Allora*

$$uV = \{(ux, m) \mid (x, m) \in V\} \tag{4.6}$$

è un test di Martin-Löf rappresentabile.

La dimostrazione si fonda sul seguente lemma.

Lemma 4.4. *Sia V un test di Martin-Löf tale che $\#(\Sigma^n \cap V_m) \leq 2^{n-m-1}$ per ogni $n \geq 0$ e $m \geq 1$. Allora V è rappresentabile.*

Sarebbe possibile indebolire l'ipotesi, richiedendo solamente $\#(\Sigma^n \cap (V_m \setminus V_{m+1})) \leq 2^{n-m-1}$, a patto di aggiungere condizioni sulla calcolabilità di $m_V(x)$. Si veda [8] a riguardo. Ma, per l'uso che ne faremo noi, non possiamo permettercelo.

Nonostante la dimostrazione sia un po' tecnica, l'idea alla base è semplice. Ricordiamo che il nostro obiettivo è costruire una $\varphi : \Sigma^* \times \mathbb{N} \to \Sigma^*$ parziale calcolabile tale che $V = V(\varphi)$, che equivale a dire

$$(x, m) \in V \iff \exists\, z \in \Sigma^* : \varphi(z, |x|) = x \wedge |z| < |x| - m$$

Doteremo la nostra φ di una proprietà più forte: la costruiremo in modo tale che $\operatorname{Im}\varphi = V_1$ e che per ogni $x \in \operatorname{Im}\varphi$ e ogni $m \in \mathbb{N}$, $1 \leq m \leq m_V(x)$, esista e sia unico uno $z \in \Sigma^*$ tale che $\varphi(z, |x|) = x$ e $|z| = |x| - m - 1$. Equivalentemente: φ deve essere iniettiva e, per ogni $n, k \in \mathbb{N}$, deve essere $\varphi(\Sigma^k \times \{n\}) = \Sigma^n \cap V_{n-k-1}$. L'ipotesi del lemma è condizione necessaria e sufficiente affinché ciò sia possibile. Vediamo ora in dettaglio come costruire la φ.

Dimostrazione. Poiché V è ricorsivamente enumerabile, esiste una funzione $g : \mathbb{N}_+ \to V$ iniettiva parziale calcolabile (nel senso che se V è finito allora è definita solo su $\{1, \ldots, |V|\}$ altrimenti è totale) che lo enumera. Poniamo $(x_i, m_i) = g(i)$.

Definiamo ora una $\varphi : \Sigma^* \times \mathbb{N} \to \Sigma^*$ parziale calcolabile dove $\varphi(z, n)$ si ottiene calcolando $i = \operatorname{str}^{-1}(z) - 2^{|z|} + 2$ e poi cercando

$$t = \min\{r \mid |x_r| = n, m_r = n - |z| - 1, \\ \#\{j \mid 1 \leq j \leq r, |x_j| = |x_r|, m_j = m_r\} = i\}$$

se lo si trova si restituisce x_t, altrimenti si diverge. Mostriamo che $V = V(\varphi)$.

Sia $(x, m) \in V$, quindi $(x, m) = (x_t, m_t) = g(t)$ per un certo $t \geq 1$. Poniamo $i = \#\{j \mid 1 \leq j \leq t, |x_j| = |x|, m_j = m\}$ e, dall'ipotesi, viene $1 \leq i \leq 2^{|x|-m-1}$. Ponendo $z = \mathrm{str}(2^{|x|-m-1} + i - 2)$ notiamo che $|z| = |x| - m - 1$ e quindi $\varphi(z, |x|) = x$ per costruzione di φ. Dunque $(x, m) \in V(\varphi)$.

Viceversa, sia $(x, m) \in V(\varphi)$ allora, per definizione, esiste $z \in \Sigma^*$ tale che $\varphi(z, |x|) = x$ e $|z| \leq |x| - m - 1$ e quindi, per costruzione di φ, esiste $r \geq 1$ tale che $x_r = x$ e $m_r = |x_r| - |z| - 1 \geq |x_r| - |x| + m + 1 - 1 = m$. Dunque $x = x_r \in V_{m_r} \subseteq V_m$, cioè $(x, m) \in V$. □

Il teorema ora segue banalmente.

Dimostrazione del teorema 4.3. Mostriamo che uV verifica l'ipotesi del lemma:

$$\#(\Sigma^n \cap (uV)_m) = \#(\Sigma^{n-|u|} \cap V_m) \leq 2^{n-|u|-m} \leq 2^{n-m-1}$$ □

4.1.2 Universalità

Definizione 4.5. Un test di Martin-Löf $\mathcal{U}$ si dice **universale** se per ogni test di Martin-Löf V vale

$$m_V(x) \leq m_{\mathcal{U}}(x) + O(1) \tag{4.7}$$

Lemma 4.5. *Se $u \in \Sigma^*$ allora si ha*

$$K(x/|x|) \leq K(ux/|ux|) + O(1) \tag{4.8}$$

Ricordiamo che con ψ denotiamo una funzione parziale calcolabile universale fissata.

Dimostrazione. Costruiamo $\varphi(z, n) = h_u(\psi(z, |u| + n))$ dove $h_u(ux) = x$ e $h_u(v) \downarrow$ se u non è prefisso di v. Quindi $\varphi(z, |x|) = x$ se e solo se $\psi(z, |ux|) = ux$.

Ne viene, per il teorema di invarianza 2.2

$$K(x/|x|) \leq K_\varphi(x/|x|) + O(1) = K(ux/|ux|) + O(1)$$ □

Teorema 4.6. *$V(\psi)$ è un test di Martin-Löf universale.*

Dimostrazione. Sia V un test di Martin-Löf e sia $a \in \Sigma$. Allora, per il teorema 4.3, $W = aV$ è rappresentabile, cioè esiste una $\varphi : \Sigma^* \times \mathbb{N} \to \Sigma^*$ parziale calcolabile tale che $W = V(\varphi)$. Quindi $m_V(x) = m_{V(\varphi)}(ax)$.

Dal teorema di invarianza 2.2 e dal lemma 4.5 sappiamo che vale

$$K(x/|x|) \leq K(ax/|ax|) + O(1) \leq K_\varphi(ax/|ax|) + O(1)$$

Supponiamo $m_V(x) > 0$. Allora

$$\begin{aligned} m_V(x) &= m_{V(\varphi)}(ax) \\ &= |ax| - K_\varphi(ax/|ax|) - 1 \\ &\leq |x| + 1 - K_\psi(x/|x|) + O(1) - 1 \\ &\leq m_{V(\psi)}(x) + O(1) + 1 \end{aligned}$$

Questa disuguaglianza vale anche nel caso $m_V(x) = 0$ (poiché $m_{V(\psi)}(x) \geq 0$). Ne viene che $V(\psi)$ è universale. □

Siamo ora in grado di ottenere il risultato fondamentale di questo capitolo.

Teorema 4.7. *Fissiamo $t \in \mathbb{N}$. Tutte le stringhe di NC_t (tranne un numero finito) sono riconosciute come casuali da ogni test di Martin-Löf.*

Dimostrazione. Sia V un test di Martin-Löf. Vogliamo mostrare che esiste un k tale che tutte le stringhe di NC_t sono $(t + k)$-casuali per V.

Dal teorema 4.6 sappiamo che $V(\psi)$ è universale e quindi vale (4.7). Per definizione di NC_t vale anche (3.1). Supponiamo $m_{V(\psi)}(x) > 0$. Allora

$$\begin{aligned} m_V(x) &\leq m_{V(\psi)}(x) + k \\ &= |x| - K(x/|x|) - 1 + k \\ &\leq |x| - |x| + t - 1 + k \\ &< t + k \end{aligned}$$

La disuguaglianza vale anche se $m_{V(\psi)}(x) = 0$ e quindi vale per ogni $x \in \Sigma^*$.

Quindi tutte le stringhe di NC_t sono $(t + k)$-casuali per V tranne quelle di lunghezza inferiore a $t + k$. □

4.2 Normalità di Borel

In questo capitolo ci limiteremo ad un solo aspetto della stocasticità appena dimostrata e lo analizzeremo in maggiore dettaglio per ottenerne conseguenze più forti. Considereremo la proprietà di una stringa di contenere in proporzioni circa uguali tutte le possibili sottostringhe di lunghezza ragionevole.

Denotiamo, come prima, $N_i(x)$ il numero di occorrenze di $i \in \Sigma$ in $x \in \Sigma^*$. Generalizziamolo definendo $N_i^m(x)$, con $m \in \mathbb{N}_+$ e $i \in \Sigma^m$, nel seguente modo: dividiamo x in blocchi di m caratteri, scartando gli eventuali caratteri in eccesso in coda, e contiamo il numero di occorrenze di i. Inoltre, denotiamo con $|x|_m$ il numero di tali blocchi (cioè $|x|_m = \left\lfloor \frac{|x|}{m} \right\rfloor$).

Definizione 4.6. Sia $\varepsilon > 0$ reale e $m \geq 1$ naturale. Una stringa $x \in \Sigma^*$, $|x| \geq 1$, si dice (ε, m)**-limitata** se per ogni $i \in \Sigma^m$ vale

$$\left| \frac{N_i^m(x)}{|x|_m} - \frac{1}{2^m} \right| \leq \varepsilon \tag{4.9}$$

Inoltre, x si dice **normale di Borel** se è $\left(\sqrt{\log|x|/|x|}, m\right)$-limitata per ogni m tale che $1 \leq m \leq \log\log|x|$.

Ogni $x \in \Sigma^*$ con $|x| < 16$ è normale di Borel poiché vale $\sqrt{\log|x|/|x|} \geq 1/2$. Invece per $|x| \geq 16$ la stringa formata da soli zeri (o da soli uni) non è normale di Borel.

Osservazione. Siano $n \geq 0$ naturale e $x > 0$ reale. È facile verificare che

$$\sum_{k=0}^{n} \binom{n}{k} (k - nx)^2 x^k (1-x)^{n-k} = nx(1-x) \tag{4.10}$$

da cui, posto $x = 1/y$, otteniamo

$$\sum_{k=0}^{n} \binom{n}{k} \left(\frac{k}{n} - \frac{1}{y}\right)^2 (y-1)^{n-k} = \frac{1}{n}(y-1)(y^{n-2}) \tag{4.11}$$

Lemma 4.8. *Siano i, m e M naturali tali che $1 \leq m \leq M$ e $1 \leq i \leq 2^m$. Sia $\varepsilon > 0$ reale. Allora*

$$\# \left\{ x \in \Sigma^M \mid \left| \frac{N_i^m(x)}{|x|_m} - \frac{1}{2^m} \right| > \varepsilon \right\} < \frac{2^{M-2m}(2^m - 1)}{\varepsilon^2 \left\lfloor \frac{M}{m} \right\rfloor} \tag{4.12}$$

Dimostrazione. Poniamo

$$T = \left\{ k \in \mathbb{N} \mid 0 \leq k \leq \left\lfloor \frac{M}{m} \right\rfloor, \left| \frac{k}{\left\lfloor \frac{M}{m} \right\rfloor} - \frac{1}{2^m} \right| > \varepsilon \right\}$$

Sostituendo $y = 2^m$ e $n = \lfloor M/m \rfloor$ in (4.11) otteniamo

$$\sum_{k=0}^{\left\lfloor \frac{M}{m} \right\rfloor} \binom{\left\lfloor \frac{M}{m} \right\rfloor}{k} \left(\frac{k}{\left\lfloor \frac{M}{m} \right\rfloor} - \frac{1}{2^m} \right)^2 (2^m - 1)^{\left\lfloor \frac{M}{m} \right\rfloor - k} = \frac{(2^m - 1)(2^{m\left\lfloor \frac{M}{m} \right\rfloor - 2m})}{\left\lfloor \frac{M}{m} \right\rfloor}$$

da cui

$$\sum_{k \in T} \binom{\left\lfloor \frac{M}{m} \right\rfloor}{k} (2^m - 1)^{\left\lfloor \frac{M}{m} \right\rfloor - k} < \frac{(2^m - 1)(2^{m\left\lfloor \frac{M}{m} \right\rfloor - 2m})}{\varepsilon^2 \left\lfloor \frac{M}{m} \right\rfloor}$$

E quindi, ricordando che $0 \leq N_i^m(x) \leq \lfloor M/m \rfloor$ e che $M = m \lfloor M/m \rfloor + \mathrm{rem}(M, m)$, risulta

$$\begin{aligned}
&\# \left\{ x \in \Sigma^M \mid \left| \frac{N_i^m(x)}{\left\lfloor \frac{M}{m} \right\rfloor} - \frac{1}{2^m} \right| > \varepsilon \right\} \\
&\quad = \sum_{k \in T} \# \left\{ x \in \Sigma^M \mid N_i^m(x) = k \right\} \\
&\quad = 2^{\mathrm{rem}(M,m)} \sum_{k \in T} \# \left\{ x \in \Sigma^{m\left\lfloor \frac{M}{m} \right\rfloor} \mid N_i^m(x) = k \right\} \\
&\quad = 2^{\mathrm{rem}(M,m)} \sum_{k \in T} \binom{\left\lfloor \frac{M}{m} \right\rfloor}{k} (2^m - 1)^{\left\lfloor \frac{M}{m} \right\rfloor - k} \\
&\quad < \frac{(2^m - 1)(2^{M-2m})}{\varepsilon^2 \left\lfloor \frac{M}{m} \right\rfloor}
\end{aligned}$$

□

Lemma 4.9. *Esiste N naturale tale che per ogni $M \geq N$ naturale si ha*

$$\# \left\{ x \in \Sigma^M \mid x \text{ non è normale di Borel} \right\} < \frac{2^M}{\sqrt{\log M}} \tag{4.13}$$

Dimostrazione. Poniamo

$$S = \{ m \in \mathbb{N} \mid 1 \leq m \leq \log \log M \}$$

Usando (4.12) con $\varepsilon = \sqrt{\log|x|/|x|}$ otteniamo

$$\begin{aligned}
&\#\left\{x \in \Sigma^M \mid x \text{ non è normale di Borel}\right\} \\
&\quad\leq \sum_{m\in S} \#\left\{x \in \Sigma^M \mid x \text{ non è } \left(\sqrt{\frac{\log|x|}{|x|}}, m\right)\text{-limitata}\right\} \\
&\quad\leq \sum_{m\in S}\sum_{i\in\Sigma^m} \#\left\{x \in \Sigma^M \mid \left|\frac{N_i^m(x)}{|x|_m} - \frac{1}{2^m}\right| > \sqrt{\frac{\log|x|}{|x|}}\right\} \\
&\quad< \sum_{m\in S} 2^m \frac{2^{M-2m}(2^m-1)}{\frac{\log M}{M}\left\lfloor\frac{M}{m}\right\rfloor} = \frac{2^M}{\log M}\sum_{m\in S}\frac{1-2^{-m}}{\frac{1}{M}\left\lfloor\frac{M}{m}\right\rfloor} \\
&\quad\leq \frac{2^M}{\log M}\sum_{m\in S} m^2(1-2^{-m}) \leq \frac{2^M}{\log M}(\log\log M)^3 \\
&\quad\leq \frac{2^M}{\sqrt{\log M}}
\end{aligned}$$

□

Vediamo ora un primo teorema che inizia a correlare la normalità di Borel con la nostra concezione algoritmica di casualità. Il teorema dice che la stringhe non normali presentano sufficiente struttura e regolarità da poter essere compresse, e il guadagno è una funzione crescente (per quanto lenta). Noi però avevamo definito le stringhe casuali come quelle in cui la compressione portava ad una riduzione della dimensione limitata da una costante. Quindi le stringhe non normali sufficientemente lunghe sono comprimibili.

Teorema 4.10 (Calude)**.** *Esistono N e c naturali tali che per ogni $x \in \Sigma^*$ con $|x| \geq N$ che non è normale di Borel vale la disuguaglianza*

$$K(x/|x|) \leq |x| - \frac{1}{2}\log\log|x| + c \tag{4.14}$$

Dimostrazione. Definiamo una $\varphi : \Sigma^* \times \mathbb{N} \to \Sigma^*$ parziale calcolabile che, su input $(\mathrm{str}(t), M)$ opera nel seguente modo: genera una ad una le $x \in \Sigma^M$, verifica la normalità di Borel e restituisce la t-esima che risulta essere non-normale, se esiste, o diverge.

Sia N ottenuto dal corollario precedente. Posto $x = f(t, M)$, con $M \geq N$, si ha $|x| = M$ e da (4.13) risulta

$$t \leq \frac{2^M}{\sqrt{\log M}} = \frac{2^{|x|}}{\sqrt{\log|x|}}$$

Per il teorema di invarianza 2.2 esiste una costante $b \geq 0$ tale che

$$\begin{aligned} K(x/|x|) &\leq K_\varphi(x/|x|) + b \\ &\leq |\operatorname{str}(t)| + b \\ &\leq \log(t+1) + b \\ &\leq \log\left(\frac{2^{|x|}}{\sqrt{\log|x|}} + 1\right) + b \\ &\leq |x| - \frac{1}{2}\log\log|x| + b + 1 \end{aligned}$$

□

Formalizziamo ora l'intuizione che avevamo espresso prima, per cui ogni stringa non normale sufficientemente lunga è comprimibile, nella sua forma inversa: ogni stringa incomprimibile sufficientemente lunga è normale di Borel.

Teorema 4.11. *Sia $t \geq 0$ naturale. Allora esiste un N_t naturale tale che ogni $x \in \mathrm{NC}_t$ con $|x| \geq N_t$ è normale di Borel.*

Dimostrazione. Siano N e c le costanti del teorema precedente. Poniamo

$$N_t = \max(N, 2^{2^{2(t+c)}})$$

Prendiamo quindi $x \in \mathrm{NC}_t$ con $|x| \geq N_t$. Se, per assurdo, non fosse normale di Borel varrebbe (4.14), cioè

$$\begin{aligned} K(x/|x|) &\leq |x| - \frac{1}{2}\log\log|x| + c \\ &\leq |x| - \frac{1}{2}\log\log 2^{2^{2(t+c)}} + c \\ &= |x| - t \end{aligned}$$

contraddizione. □

Vediamo ora un ultimo teorema molto interessante: ogni stringa può essere inclusa in una stringa t-incomprimibile, cioè per ogni stringa x esiste una stringa t-incomprimibile w tale che x è sottostringa di w.

Teorema 4.12. *Sia $t \geq 0$ naturale. Allora per ogni $x \in \Sigma^*$ esistono $u, v \in \Sigma^*$ tali che $uxv \in \mathrm{NC}_t$.*

Dimostrazione. Sia $x \in \Sigma^*$, $|x| = n$. Scegliamo $z \in \mathrm{NC}_t$ tale che $|z| \geq 2^{2^{2n+1}}$ e $|z| \geq N_t$. Vogliamo mostrare che x compare come sottostringa in z. Anzi, di più: che $N_x^n(z) > 0$. Per il teorema precedente sappiamo che z è normale di Borel, cioè

$$\left| \frac{N_i^m(z)}{|z|_m} - \frac{1}{2^m} \right| \leq \sqrt{\frac{\log |z|}{|z|}}$$

per ogni $1 \leq m \leq \log\log |z|$ e ogni $i \in \Sigma^m$. In particolare vale per $m = n$ e $i = x$, e quindi ne viene

$$\frac{N_x^n(z)}{|z|_n} \geq \frac{1}{2^n} - \sqrt{\frac{\log |z|}{|z|}}$$

Ora, per avere che $N_x^n(z) > 0$ basta mostrare che $\sqrt{\log |z| / |z|} < \frac{1}{2^n}$. Ciò vale in quanto

$$\sqrt{\frac{\log |z|}{|z|}} = 2^{\frac{2n+1}{2} - \frac{2^{2n+1}}{2}} = 2^{n+\frac{1}{2}-2^{2n}} < 2^{n+\frac{1}{2}-2n-\frac{1}{2}} = 2^{-n} \qquad \square$$

Capitolo 5

Conclusioni

Nell'ultimo capitolo abbiamo osservato come la caratterizzazione della stocasticità tramite l'incomprimibilità algoritmica risulti estremamente soddisfacente: non c'è modo tramite test statistici effettivi di rilevare regolarità nelle stringhe incomprimibili e quindi esse sono indistinguibili da stringhe generate casualmente per tutti gli scopi pratici.

Il nostro è stato solo un breve assaggio dell'argomento. Ci sarebbero molti altri aspetti da esplorare. Ad esempio osservare che i programmi minimali sono incomprimibili (perché altrimenti potremmo comprimere ulteriormente anche la stringa originale) e quindi casuali. Oppure che l'insieme delle stringhe casuali è immune, nel senso della teoria della calcolabilità.

È di notevole interesse anche l'estensione di questa teoria alle stringhe infinite, ovvero alle successioni in Σ, per le quali i test di Martin-Löf hanno una definizione molto più naturale.

Bibliografia

[1] Cristian S. Calude. *Information and Randomness: An Algorithmic Perspective*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2002.

[2] Cristian S. Calude e Ion Chiţescu. «Representability of recursive P. Martin-Löf tests». In: *Kybernetika* 19.6 (1983), pp. 526–536.

[3] Thomas M. Cover e Joy A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[4] Peter Grünwald e Paul Vitányi. «Shannon Information and Kolmogorov Complexity». Ott. 2004.

[5] Andrei N. Kolmogorov. «Three Approaches to the Quantitative Definition of Information». In: *Problems of Information Transmission* 1.1 (1965), pp. 1–7.

[6] Ming Li e Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Texts in Computer Science. Springer, 2008.

[7] Per Martin-Löf. «The Definition of Random Sequences». In: *Information and control* 9 (1966), pp. 602–619.

[8] Ludwig Staiger. «Representable P. Martin-Löf tests». In: *Kybernetika* 21.3 (1985), pp. 235–243.

[9] Alexander K. Zvonkin e Leonid A. Levin. «The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms». In: *Russian Mathematical Surveys* 25.6 (1970), pp. 83–124.