Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea Magistrale in Matematica

CASUALITÀ ALGORITMICA E DIMENSIONE EFFETTIVA

Presentata da:
GUIDO FIORILLO

Relatore:
Chiar.mo Prof.
GUIDO GHERARDI

Correlatore:
Chiar.mo Prof.
MARCO LENCI

♯ Anno accademico
2019/2020

Indice

Introduzione **5**

1 Misure sullo spazio di Cantor **11**
1.1 Primo metodo di costruzione di una misura 13
1.2 Secondo metodo di costruzione di una misura 14
1.3 Richiami di teoria della computabilità 16
1.4 Complessità a priori 17

2 Casualità algoritmica **27**
2.1 Il concetto di casualità algoritmica 27
2.2 Classi Π^0_1 di misura positiva 35
2.3 Il teorema di Kučera 36
2.4 Il concetto di stocasticità secondo von Mises-Church 39

3 Dimensione di Hausdorff effettiva **45**
3.1 Casualità per la misura di Hausdorff 45
3.2 Dimensione di Hausdorff e complessità 51

Introduzione

Quando si può affermare che una sequenza di 0 e di 1 è casuale? Ci sono sequenze che nessuno definirebbe casuali, come:

$$0101010101010101010101\ldots$$

oppure

$$010010001000010000001\ldots$$

mentre altre paiono essere tali, come:

$$0101101011010001110 01\ldots$$

ma nessuna risposta è davvero sensata finché non si è stabilito che cosa intendiamo per casualità. Partendo dagli esempi appena fatti, si possono individuare tre proprietà che caratterizzano le sequenze casuali in senso intuitivo:

1. *Una sequenza casuale dev'essere priva di struttura.*
   Ancora una volta, la nozione di "struttura" è piuttosto vaga, ma possiamo precisarla meglio ricorrendo all'idea di *una descrizione della sequenza che contiene tutte le informazioni necessarie a ricostruirla.* Nel primo esempio che abbiamo fatto, una simile descrizione si può dare in modo molto semplice: basta dire che è formata dalla ripetizione della stringa 01. D'altra parte sorgono due problemi: è chiaro che ogni sequenza può essere in qualche modo descritta, perché è evidente che, nel peggiore dei casi, basterà indicarne esplicitamente i segmenti iniziali. L'esistenza di una struttura sembra legata alla possibilità non tanto di indivduare una descrizione qualunque, ma di trovarne una che sia *notevolmente più corta* del segmento inziale che cerchiamo di ricostruire. Inoltre, bisognerà specificare quali operazioni sono consentite all'agente che tenta di ricostruire la sequenza; se questo fosse dotato di una certa "preveggenza" o potesse consultare un oracolo che gli suggerisce certi bit, potrebbe ricostruire la stringa pur avendone una descrizione molto breve.

2. *Se vediamo un segmento iniziale della stringa, non deve essere possibile prevedere sulla sua base se il prossimo bit sarà 0 o 1*
   In questo caso ragioniamo in termini probabilistici. Pensiamo, infatti, di avere una successione di 0 e di 1 in cui non ci sia nessuna struttura evidente e supponiamo che rappresenti gli esiti del lancio ripetuto di una moneta. Se la sequenza è davvero casuale, cioè la moneta non è truccata, la probabilità che esca 1 è pari a 1/2 indipendentemente da quanto è successo in precedenza e non è, dunque, possibile vincere somme arbitrariamente grandi scommettendo sui lanci. Al contrario, se la moneta è truccata, può darsi, ad esempio, che quando l'ultimo lancio ha prodotto uno 0, la probabilità che il prossimo dia 1 sia maggiore di un mezzo, ed è facile escogitare una strategia che permette di vincere sempre più denaro.

3. *La successione deve superare certi test statistici.* Ad esempio, in una successione veramente casuale ci aspettiamo che la frequenza di 0 e di 1 nei suoi segmenti iniziali tenda a 1/2 mano a mano che questi segmenti diventano più lunghi. Vediamo però dal nostro primo esempio che ci sono sequenze che superano tale test ma certamente non sono casuali. Sarà dunque meglio richiedere che il test venga superato da una certa classe di sottosequenze della sequenza data. Ancora una volta, è necessario specificare con cautela quale sia questa classe, perché se è troppo ampia rischiamo di concludere che *nessuna* sequenza è casuale.

Le nozioni che abbiamo appena discusso sono evidentemente correlate fra loro, ma è difficile capire in che modo finché non se ne danno definizioni esatte. L'obiettivo di questa tesi è mostrare gli strumenti matematici con cui si può formalizzare in modo rigoroso la nozione di casualità. Nonostante la precisione dei concetti matematici, rimane pur sempre a noi la responsabilità di decidere quale sia l'idea intuitiva più adatta a catturare la nozione di casualità e scelte diverse condurranno a diverse definizioni. D'altra parte, la formalizzazione porta a comprendere che alcune idee in apparenza diverse sono in realtà equivalenti; se si trova un concetto che riunisce in sé varie delle nostre idee intutitive, si tratterà di una nozione particolarmente significativa. Il primo approccio matematico al problema fu tentato da R. von Mises nel 1919. Egli, seguendo l'idea esposta nel punto 3, definì casuali quelle sequenze tali che ogni loro sottosequenza estratta in modo ammissibile rispetti la legge dei grandi numeri. Il problema, come già fatto notare, sta tutto nello specificare quali siano le procedure di estrazione ammissibili. Von Mises propose che la selezione avvenisse tramite una sorta di gioco: un avversario ci sottopone una sequenza Z e ci sfida a trovarne una sottosequenza Z' che violi la legge dei grandi numeri. All' $n-$esimo turno, l'avversario ci svela i primi

n bit di Z e, sulla base di questi, noi dobbiamo decidere se inserire o no in Z' il bit di posto $n+1$, dopodiché questo viene svelato e si passa al turno successivo. In realtà, se non si pongono limiti sulle strategie lecite, ci è possibile scegliere una sottosequenza qualunque. Ad esempio, se disponiamo *a priori* di una lista che indica la posizione di tutti gli 0, possiamo facilmente estrarre una sottosequenza per cui la frequenza relativa degli 0 è identicamente nulla. Una restrizione sostanziale delle strategie ammissbili venne introdotta solo nel 1940 da A. Church, che nota come, per un giocatore che voglia vincere, non sia di nessun aiuto avere una strategia che corrisponde a una funzione ben definita ma priva di una procedura effettiva per calcolarne i valori. Le funzioni (cioè le strategie) che possono essere effettivamente calcolate corrispondono, secondo la famosa tesi di Church-Turing, a quelle calcolabili da una macchina di Turing. Church propone, pertanto, di limitare la scelta delle strategie a quest'ultima classe e ottiene così una definizione non banale di casualità, oggi nota come stocasticità di Von Mises-Church.
D'altra parte, già nel 1939 Ville aveva dimostrato un risultato che mette in discussione l'appropriatezza della definizione di von Mises-Church. Il problema è che esiste una sequenza che supera il test di Von Mises-Church, ma in cui la frequenza relativa degli 1 è costantemente inferiore a quella degli 0, cioè:

$$\frac{|\{k \leq n \,:\, X(k) = 1\}|}{n} < \frac{1}{2}$$

per ogni n. Ora, è controintutivo che una simile sequenza sia casuale; se lo fosse, ci aspetteremmo che la frequenza relativa degli 1 tendesse a 1/2 oscillandovi intorno, e non restando sempre al di sotto. Ville propose di aggiungere alla legge dei grandi numeri anche un altro test statistico, e cioè la legge del logartimo iterato, che permette di escludere la sequenza da lui costruita. Si tratta, però, di un espediente insoddisfacente, perché è chiaro che potrebbe esserci un'altra sequenza che sfugge anche a quest'ulteriore test. L'unica vera soluzione è quella di introdurre un metodo di generare sequenza che sfuggano a ogni possibile test statistico.
Ciò venne fatto da P. Martin-Löf nel 1966, tramite una versione effetiva del concetto, classico in teoria della misura, di "insieme di misura nulla". Classicamente, un sottoinsieme A della retta reale è di misura nulla se per ogni n naturale esiste una famiglia $\{I_k^n\}_{k\in\mathbb{N}}$ di intervalli aperti che ricoprono A e tali che $\sum_k l(I_k^n) \leq \frac{1}{2^n}$, dove $l(I)$ indica la lunghezza di un intervallo. A meno di identificare le sequenze di bit con numeri reali fra 0 e 1 (le difficoltà che insorgono a questo proposito per ora non ci interessano), si può trasferire questo concetto sullo spazio di tutte le sequenze infinite di bit. Ora la nozione può essere resa effettiva richiedendo che gli intervalli I_k^n siano computabilmente enumerabili in modo uniforme, cioè deve esistere una macchina

di Turing che, dati n e k, individua l'intervallo I_k^n (cioè calcola una stringa che codifica questo intervallo). Ogni successione $\{\bigcup_k I_k^n\}_{n\in\mathbb{N}}$ di ricoprimenti aperti effettivi rappresenta allora un test statistico e una sequenza si dice casuale se supera tutti questi test (cioè non sta nell'intersezione $\bigcap_n\{\bigcup_k I_k^n\}$). Questa definizione è molto efficace: implica, infatti, la stocasticità di Von Mises-Church, ma non può essere elusa allo stesso modo. Inoltre, si dimostra che è equivalente a nozioni che formalizzano i punti 1 e 2 della nostra lista intuitiva di requisiti per una sequenza casuale.

1. L'idea di "descrizione della sequenza che contiene tutte le informazioni necessarie a ricostruirla" può essere catturata dalla nozione di *complessità di Kolmogorov* K, che risulta essere, intutitivamente, la lunghezza della più corta di queste descrizioni. Si dimostra che Z è casuale nel senso di Martin-Löf se e solo se per i suoi segmenti iniziali $Z|_n$ vale:

   $$K(Z|_n) \geq n - O(1)$$

   cioè se la descrizione più corta di $Z|_n$ non è (per n grande) significativamente più corta di $Z|_n$.

2. L'idea di strategia in un gioco equo basato sui bit di Z, interpretati come "testa" e "croce" di una moneta, è formalizzata dal concetto di *martingala inferiormente semicomputabile*. Si tratta di una funzione definita sulle stringhe finite e che assume valori reali. Se una certa stringa σ rappresenta gli esiti del lancio della moneta, $f(\sigma)$ sarà il capitale che possediamo se scommettiamo su questi lanci seguendo la nostra strategia. Si vede che Z è casuale nel senso di Martin-Löf se e solo se nessuna di tali martingale assume valori arbitrariamente grandi sui segmenti iniziali di Z.

La nozione di casualità di Martin-Löf include dunque molte idee intutitive sulla casualità. Uno dei risultati fondamentali della teoria è l'esistenza di un test *universale*; se una sequenza supera quest'ultimo, supera certamente anche tutti gli altri, per cui è casuale. Le sequenze Martin-Löf-casuali, inoltre, hanno proprietà interessanti dal punto di vista computazionale.
Rimane, in ogni caso, una questione aperta. Supponiamo di avere una sequenza casuale Z data da:

$$z_0z_1z_2z_3z_4z_5z_6\ldots$$

e di ripetere i suoi bit, in modo da ottenere una sequenza Z':

$$z_0z_0z_1z_1z_2z_2z_3z_3,\ldots$$

opppure di interporre degli zeri e ottenere la sequenza Z'':

$$z_0 0 z_1 0 z_2 0 z_3 0 \dots$$

Z' e Z'' non sono più casuali nel senso di Martin-Löf, perché, ad esempio, si vede subito che la loro complessità di Kolmogorov di un loro segmento iniziale di lunghezza n è circa $n/2$; d'altra parte, è ragionevole supporre che continuino a contenere una certa casualità e sembra desiderabile riuscire a quantificarla, dicendo ad esempio che queste sequenze sono "casuali a metà", mentre:

$$Z''' = z_1 00 z_2 00 z_3 00 \dots$$

è casuale "per un terzo" e

$$Z'''' z_1 000000 \dots$$

non è casuale per nulla.
Poiché la nozione di casualità di Martin-Löf viene dalla teoria della misura sui reali, pare proficuo sfruttare ancora l'analogia. Nel caso di un insieme di misura nulla sui reali, come ad esempio la cosiddetta "polvere di Cantor", per indagarne ulteriormente la struttura si può ricorrere alla misura di Hausdorff e definirne così la dimensione di Hausdorff. Nel 2000, J. Lutz ha sviluppato una nozione effettiva di dimensione di Hausdorff e, anche se il suo approccio originale usava le supermartingale, è possibile definire questo concetto in modo del tutto parallelo a quanto fatto classicamente in analisi: in primo luogo si definiscono gli "insiemi di misura effettiva $s-$dimensionale nulla" e poi, data una successione A si pone la sua dimensione effettiva uguale all'estremo inferiore degli s per cui la misura effettiva $s-$dimensionale di $\{A\}$ è 0. Per quanto la procedura sia tecnicamente più complessa di quella seguita per la casualità di Martin-Löf, si trova infine un risultato che lega profondamente la dimensione effettiva all'idea informale di una successione in parte casuale:

$$\dim_H^1(A) = \liminf \frac{K(A|_n)}{n},$$

dove $A|_n$ indica il segmento iniziale di lunghezza n di A. Ciò permette,ad esempio, di affermare rigorosamente che gli insiemi Z' e Z'' hanno dimensione effettiva $1/2$, Z''' ha dimensione $1/3$ e Z'''' ha dimensione 0.
Nel capitolo 1 della tesi introduciamo dapprima lo "spazio di Cantor", uno spazio topologico che contiene tutte le successioni infinite di bit. Affrontiamo poi alcuni concetti classici di teoria della misura, individuando, in particolare, due modalità di introdurre una misura sullo spazio di Cantor. Il primo, basato sul teorema di estensione di Carathéodory, porta a definire la misura

di Lebesgue, mentre il secondo, che deriva dalla possibilità di vedere lo spazio di Cantor come spazio metrico, permette di definire le misure e la dimensione di Hausdorff. In seguito richiamiamo alcune idee di teoria della computabilità e caratterizziamo le misure immagine della misura di Lebesgue tramite un funzionale di Turing. Questa discussione ci porta a definire la "complessità a priori" di una stringa.
Nel secondo capitolo presentiamo la teoria della casualità di Martin-Löf, presentandone tre caratterizzazioni equivalenti che corrispondono alle idee intutitive sulla casualità descritte in precedenza. Passiamo poi a esaminare le proprietà computazionali delle successioni casuali. Dimostriamo, in primo luogo che ogni classe effettivamente chiusa (cioé Π^0_1) contiene rappresentanti per tutti i gradi di Turing di sequenze casuali. In seguito, descriviamo una procedura uniforme che permette di codificare una successione qualunque in una successione casuale (teorema di Kučera). Infine, introduciamo in modo rigoroso la nozione di stocasticità secondo von-Mises-Church e mostriamo che tutte le sequenze casuali nel senso di Martin-Löf sono stocastiche in questa accezione.
Nel terzo capitolo introduciamo il concetto di dimensione effettiva e casualità rispetto alla misura di Hausdorff. Proviamo a riproporre caratterizzazioni analoghe a quelle date per la casualità di Martin-Löf nel capitolo 2, ma le difficoltà che insorgono ci spingono a introdurre una diversa e più forte nozione di casualità per la misura di Hausdorff. Quest'ultima si presta a essere caratterizzata tramite le supermartingale e la complessità a priori definita nel capitolo 1. In base a questi risultati, si riesce a ottenere una formula per calcolare la dimensione effettiva di una sequenza:

$$\dim^1_H(A) = \liminf \frac{K(A|_n)}{n}.$$

La tesi si chiude con due esempi che mostrano che per ogni razionale r esiste una successione di dimensione effettiva r.

Capitolo 1

Misure sullo spazio di Cantor

In questo capitolo ci occupiamo delle misure definite sullo spazio di Cantor $2^{\mathbb{N}}$, introducendo nozioni che verranno spesso usate in seguito e modificate per ottenerne versioni effettive. Quanto vedremo per lo spazio di Cantor non è altro che l'applicazione di risultati generali di teoria della misura, quali sono contenuti, ad esempio nei libri [2] e [7], nonché nell'articolo [8] cui faremo spesso riferimento.
Fissiamo, in primo luogo, alcune notazioni. Nel seguito, parleremo spesso di stringhe finite di bit; l'insieme che le contiene tutte verrà indicato con $2^{<\omega}$, mentre la famiglia delle stringhe di lunghezza esattamente n verrà indicata con 2^n. Sull'insieme delle stringhe finite si possono definire varie relazioni d'ordine; quelle che noi useremo saranno l'ordinamento lessicografico, che è totale, e l'ordinamento per prefisso, per cui useremo il simbolo $\preceq$:

$$\sigma \preceq \tau \quad \Leftrightarrow \quad \sigma \text{ è prefisso di } \tau.$$

Tale relazione d'ordine non è totale; quando due stringhe non sono confrontabili, cioé nessuna delle due è prefisso dell'altra, diremo spesso che sono incompatibili. Considereremo anche l'insieme delle successioni infinite di bit; questo verrà detto spazio di Cantor e indicato con la notazione $2^{\mathbb{N}}$. Questo insieme è dotato della topologia prodotto di quella discreta su ciascun fattore $2 = \{0, 1\}$. L'ordinamento per prefisso si estende in modo ovvio a $2^{<\omega} \cup 2^{\mathbb{N}}$. Data una stringa σ, si può definire allora la sua *foresta*, cioè l'insieme delle successioni infinite di bit che cominciano con σ:

$$[\![\sigma]\!] = \{X \in 2^{\mathbb{N}} \,:\, \sigma \preceq X\}$$

Se R è un insieme di stringhe, indicheremo con $[\![R]\!]$ l'unione delle foreste delle stringhe appartenenti a R: La topologia dello spazio di Cantor ammette una base data dalla famiglia di aperti $\{[\![\sigma]\!] \,:\, \sigma \in 2^{<\omega}\}$ ed è metrizzabile; una metrica che la induce si può definire come segue:

$$d(X,Y) = \begin{cases} 0 & \text{se} X = Y \\ \dfrac{1}{2^n} & \text{se n è il minimo dei } k \text{ per cui } X(k) \neq Y(k) \end{cases}$$

Allora, l'insieme $[\![\sigma]\!]$ si può vedere come la palla di raggio $2^{-|\sigma|}$ centrata in qualunque $X \succ \sigma$. Scegliendo per ogni σ un $X_\sigma \succ \sigma$ si ottiene un sottoinsieme denso e numerabile dello spazio di Cantor.
Ricordiamo che un'algebra su un insieme A è una famiglia $\mathfrak{A}$ di sottoinsiemi di A tali che:

1. $\emptyset, A \in \mathfrak{A}$
2. Se $B \in \mathfrak{A}$, $B^c \in \mathfrak{A}$
3. Se $B_1, B_2, \dots B_n \in \mathfrak{A}$, $\bigcup_{j=1}^n B_j \in \mathfrak{A}$ e $\bigcap_{j=1}^n B_j \in \mathfrak{A}$

Se oltre a 3 vale anche:

4. Se $B_1, B_2, \dots \in \mathfrak{A}$, $\bigcup_{j=1}^\infty B_j \in \mathfrak{A}$ e $\bigcap_{j=1}^\infty B_j \in \mathfrak{A}$

si dice che $\mathfrak{A}$ è una $\sigma-$algebra.
Se $\mathfrak{S}$ è una famiglia di sottoinsiemi di A che soddisfa le proprietà seguenti:

1. $\emptyset \in \mathfrak{S}$
2. $S, T \in \mathfrak{S}$ implica $S \cap T \in \mathfrak{S}$
3. se $S, T \in \mathfrak{S}$, esistono $S_1, S_2, \dots, S_n \in \mathfrak{S}$ disgiunti tali che:

$$S/T = \bigcup_{j=1}^{n} S_j$$

$\mathfrak{S}$ si dice essere un *semianello*.

Chiameremo $\mathfrak{C}$ la $\sigma-$algebra generate dalla famiglia $\mathfrak{C}_0 = \{[\![\sigma]\!] : \sigma \in 2^{<\omega}\} \cup \{\emptyset\}$. Si osservi che $\mathfrak{C}_0$ è un semianello. Siccome lo spazio ha una base numerabile, è immediato dedurre che $\mathfrak{C}$ contiene la $\sigma-$algebra dei boreliani.
Una funzione μ^* a valori in $\mathbb{R} \cup \{+\infty\}$ definita sui sottoinsiemi di A e tale che:

1. $\mu^*(H) \geq 0$ per ogni $H \subseteq A$
2. $\mu^*(H) \leq \mu^*(H')$ se $H \subseteq H'$

3. Se $H_1, H_2, \dots \subseteq A$ sono a dua a due disgiunti,
$$\mu^*\left(\bigcup_{j=1}^{\infty} H_j\right) \leq \sum_{j=1}^{\infty} \mu^*(H_j)$$

si dice *misura esterna* su A.
Se μ è una funzione a valori reali definita su una $\sigma-$algebra $\mathfrak{A}$ e verifica le condizioni

1. $\mu^*(H) \geq 0$ per ogni $H \in \mathfrak{A}$
2. $\mu^*(H) \leq \mu^*(H')$ se $H, H' \in \mathfrak{A}$ e $H \subseteq H'$
3. Se $H_1, H_2, \dots \in \mathfrak{A}$ sono a due a due disgiunti,
$$\mu\left(\bigcup_{j=1}^{\infty} H_j\right) = \sum_{j=1}^{\infty} \mu(H_j)$$

μ si dice una *misura* su $\mathfrak{A}$.
Infine, se μ è definita su un semianello $\mathfrak{S}$ e valgono le condizioni:

1. $\mu^*(H) \geq 0$ per ogni $H \in \mathfrak{S}$
2. $\mu^*(H) \leq \mu^*(H')$ se $H, H' \in \mathfrak{S}$ e $H \subseteq H'$
3. Se $H_1, H_2, \dots \in \mathfrak{S}$ sono a due a due disgiunti e $\bigcup_{j=1}^{\infty} H_j \in \mathfrak{S}$,
$$\mu\left(\bigcup_{j=1}^{\infty} H_j\right) = \sum_{j=1}^{n} \mu(H_j)$$

μ si chiama misura (sul semianello $\mathfrak{S}$)

1.1 Primo metodo di costruzione di una misura

Definizione 1.1.1. Una *premisura* è una funzione $\rho : 2^{<\omega} \longrightarrow \mathbb{R}^{\geq 0}$.

Teorema 1.1.2. *Se ρ è una premisura,*
$$\mu^*(C) = \inf\{\sum_{\sigma \in R} \rho(\sigma) \ : \ C \subseteq [\![R]\!]\} \tag{1.1}$$

è una misura esterna sullo spazio di Cantor.

L'enunciato precedente è il teorema 4 a pag. 9 di [2], riscritto con una diversa terminologia.

Definizione 1.1.3 (Condizione di Carathéodory)**.** La classe $C \subseteq 2^{\mathbb{N}}$ si dice μ^*-*misurabile* se vale la condizione seguente:

$$\mu^*(E) = \mu^*(E \cap C) + \mu^*(E \cap C^c) \tag{1.2}$$

per ogni $E \subseteq 2^{\mathbb{N}}$.

Teorema 1.1.4. *La famiglia $\mathfrak{A}$ delle classi μ^*-misurabili è una σ-algebra. Se μ è la restrizione di μ^* a $\mathfrak{A}$, μ è una misura.*

Dimostrazione. Si guardi il Teorema 1.3 in [2] □

Una misura si dice *finita* se assume valori reali (cioé non è mai infinita).

Teorema 1.1.5 (Teorema di estensione di Carathéodory)**.** *Se l è una misura finita su un semianello $\mathfrak{S}_0$, esiste un'unica misura che estende l alla σ-algebra $\mathfrak{A}$ generata da $\mathfrak{S}_0$.*

Dimostrazione. Si guardi il teorema 6.1 in [7] □

Definizione 1.1.6. Sia ρ una premisura sullo spazio di Cantor. Essa si dice *premisura di probabilità* se:

1. $\rho(\oslash) = 1$
2. $\rho(\sigma) = \rho(\sigma 0) + \rho(\sigma 1)$

La condizione 2 serve ad assicurare che ρ induca una misura sul semianello $\mathfrak{S}$. Dal teorema di estensione di Carathéodory discende che ρ si estende a una misura μ sui boreliani tale che per ogni stringa σ, $\mu([\![\sigma]\!]) = \rho(\sigma)$. Dalla condizione 1 deriva allora che $\mu(2^{\mathbb{N}}) = 1$. L'esempio più tipico di questa costruzione si ottiene considerando la premisura di probabilità $\rho(\sigma) = 2^{-|\sigma|}$; la misura da questa ottenuta si dice *misura di Lebesgue sullo spazio di Cantor*.

1.2 Secondo metodo di costruzione di una misura

Se la premisura ρ non soddisfa la condizione 2 della definizione di misura di probabilità, essa non definisce una misura sul semianello $\mathfrak{C}_0$, per cui in generale non si può usare il teorema di estensione di Carathéodory per ottenere una misura sui boreliani; se, però, ci si trova in uno spazio metrico, è possibile adottare un metodo alternativo per costruire una misura siffatta. Noi considereremo solo il caso dello spazio di Cantor; il problema nella massima generalità viene trattato nel libro [2].

Definizione 1.2.1. Dato $A \subseteq 2^{\mathbb{N}}$, un insieme di stringhe D viene detto un n-ricoprimento di A se $A \subseteq [\![D]\!]$ e inoltre per ogni $\sigma \in D$ $|\sigma| \geq n$.

Definizione 1.2.2. Una funzione $h : \mathbb{N} \mapsto \mathbb{R}^{\geq 0}$ non decrescente e illimitata si dice *funzione d'ordine*; essa induce una premisura sulle stringhe data da:

$$\rho_h(\sigma) = 2^{-h(|\sigma|)}$$

Definizione 1.2.3. Poniamo:

$$H_n^h(A) = \inf\{\sum_{\sigma \in D} \rho_h(\sigma) \ : \ D \text{ è un } n\text{-ricoprimento di } A\} \tag{1.3}$$

Dal fatto che un n-ricoprimento è anche un m-ricoprimento se $m \leq n$, si deduce che la successione $H_n^h(A)$ è non decrescente, per cui esiste il suo limite per $n \to \infty$.

Definizione 1.2.4. Si pone:

$$H^h(A) = \lim_{n \to \infty} H_n^h(A)$$

H^h di chiama misura di Hausdorff associata ad h.

Si dimostra che le misure di Hausdorff rendono misurabili tutti i boreliani. Solitamente, saremo interessati agli insiemi di misura nulla, per i quali vale il seguente:

Lemma 1.2.5. *A ha misura H^h nulla se e solo se esiste una successione di insiemi di stringhe D_n tali che per ogni n, $A \subseteq [\![D_n]\!]$ e*

$$\sum_{\sigma \in D_n} \rho_h(\sigma) \leq 2^{-n}$$

Dimostrazione. Se la classe A ha misura H^h nulla, per ogni n esiste un k tale che $H_k^h(A) \leq 2^{-(n+1)}$; allora esiste un k-ricoprimento D tale che $\sum_{\sigma \in D} \rho_h(\sigma) \leq 2^{-n}$. Viceversa, sia data una successione D_n di insiemi di stringhe con le proprietà enunciate nel lemma. Mostriamo che per ogni n è possibile trovare un k tale che $H_k^h(A) \leq 2^{-n}$ Osserviamo che se $\sigma \in D_n$, $\rho_h(\sigma) \leq 2^{-n}$, per cui $h(|\sigma|) \geq n$. Ora fissiamo k; sia n più grande di $h(1), h(2), \ldots, h(k-1)$. Allora, $\sigma \in D_n$ implica $h(|\sigma|) > h(1), h(2), \ldots, h(k-1)$, per cui $|\sigma| \geq k$ e D_n è un k-ricoprimento. $\square$

Una classe particolarmente importante di funzioni d'ordine è data da $h(n) = tn$ con $t \in [0, 1]$. Per indicare le misure di Hausdorff associate a tali funzioni, si usa la notazione H^t.

Teorema 1.2.6. *Per ogni $A \subseteq 2^{\mathbb{N}}$, esiste un unico $t \in [0, 1]$ tale che*

1. *$H^s(A) = +\infty$ se $s \in [0, t[$*
2. *$H^s(A) = 0$ se $s \in]t, 1]$*

Dimostrazione. Siano $r, s \geq 0$, $r + s \leq 1$. Allora:

$$\begin{aligned} H_n^{r+s}(A) &= \inf\{\sum_{\sigma\in D} 2^{-r|\sigma|}2^{-s|\sigma|} \ : \ D \text{ è un } n\text{-ricoprimento di } A\} \\ &\leq 2^{-ns}\inf\{\sum_{\sigma\in D} 2^{-r|\sigma|} \ : \ D \text{ è un } n - \text{ricoprimento di } A\} = 2^{-ns}H_n^r \end{aligned} \tag{1.4}$$

Da questo si deduce che se $H^r(A) < +\infty$, $H^{r+s}(A) = 0$ per tutti gli $s \geq 0$ e se esiste s $H^{r+s}(A) = \infty$, $H^r(A) = \infty$. Allora basta porre $t = \sup\{r \ : \ H^r(A) = +\infty\}$ □

Definizione 1.2.7. Sia $A \subseteq 2^{\mathbb{N}}$. Si definisce come segue la sua dimensione di Hausdorff:

$$dim_H(A) = \inf\{r \ : \ H^r(A) = 0\}$$

1.3 Richiami di teoria della computabilità

In questo paragrafo richiamiamo brevemente alcuni concetti di teoria della computabilità e fissiamo le relative notazioni.
Assumiamo per nota la nozione di macchina di Turing. Le macchine di Turing possono essere elencate in modo effettivo, per cui ha senso parlare dell'$e-$esima macchina di Turing. Ciò porta a un'enumerazione effettiva $\{\Phi_e\}_{e\in\mathbb{N}}$ delle funzioni computabili parziali. A rigore, tale funzioni vanno da $\mathbb{N}$ a $\mathbb{N}$, ma, fissando una certa codifica, si può parlare, ad esempio, di funzioni che hanno come dominio o codominio, ad esempio, l'insieme delle stringhe finite $2^{<\omega}$ o il prodotto cartesiano $\mathbb{N} \times \mathbb{N}$.
Se la funzione Φ_e è ben definita sull'argomento n, cioé se la computazione effettuata dalla corrispondente macchina di Turing sull'input n converge, scriviamo $\Phi_e(n) \downarrow$; in caso contrario si scrive $\Phi_e(n) \uparrow$. Se la e-esima macchina di Turing dopo s passi termina producendo l'output y si scrive $\Phi_{e,s}(n) \downarrow= y$; se dopo s passi la computazione eseguita dalla macchina ancora non è ancora terminata, si scrive $\Phi_{e,s}(n) \uparrow$.

Un insieme si dice *computabile* se la sua funzione caratteristica è computabile. Si dice *computabilmente enumerabile o c.e.* se è il dominio di una funzione computabile parziale. C'è dunque un'enumerazione effettiva $\{W_e\}_{e\in\mathbb{N}}$ degli insiemi c.e.:

$$W_e = \{n \,:\, \Phi_e(n)\downarrow\}.$$

L'insieme W_e ammette approssimazioni finite e computabili dal basso $W_{e,s}$:

$$W_{e,s} = \{n \,:\, \Phi_{e,s}(n)\downarrow\}$$

con la proprietà che $W_{e,s} \subseteq W_{e,s+1}$ e $W_e = \bigcup_s W_{e,s}$ Un insieme c.e. verrà anche detto Σ^0_1; un insieme il cui complementare è c.e. si dice Π^0_1.
Un insieme è computabile se e solo se sia esso sia il suo complementare sono c.e; un esempio classico di insieme c.e. ma non computabile è:

$$\emptyset' = \{e \,:\, \Phi_e(e)\uparrow\}.$$

Si può generalizzare il concetto di macchina di Turing considerando l'eventualità che questa abbia a disposizione un oracolo, cioè conosca un certo insieme X. Ciò si realizza dando alla macchina un nastro su cui è riportato un 1 in $n-$ esima posizione se e solo se $n \in X$. Il principio dell'uso afferma che ogni computazione convergente usa solo un numero finito di interrogazioni all'oracolo, per cui il suo risultato su un certo output dipende solo da un segmento iniziale di X (se questo viene interpretato come successione infinita di bit). Se l'insieme X è finito, ha senso usare come oracolo anche una stringa finita. La funzione parziale individuata dall'$e-$esima macchina di Turing con oracolo X si indica con Φ^X_e. Tali funzioni sono, intuitivamente, computabili relativamente a X, cioé chiunque conosca la funzione caratteristica dell'insieme X sa calcolare in modo effettivo anche tali funzioni.
Con abuso di notazione, diciamo che $\Phi_e(X) = Y$ se la $e-$esima macchina di Turing con oracolo X calcola la funzione caratteristica di Y. Associando a ogni X l'insieme Y (se esiste) tale che $\Phi_e(X) = Y$ si definisce una funzione parziale sullo spazio di Cantor; tali funzioni sono dette, di solito, funzionali di Turing. Nel prossimo paragrafo sarà data una caratterizzazione dei funzionali di Turing rispetto a come si comportano sui segmenti inziali di X e Y.

1.4 Complessità a priori

In questa sezione, inroduciamo il concetto di complessità a priori e ne esaminiamo i legami con la complessità di Kolmogorov, seguendo il capitolo 3 del volume [6] di Downey e Hirschfeldt. Il significato di tali concetti per la teoria della computabilità viene discusso nei capitoli 4 e 5 del volume *Kolmogorov complexity and algorithimic randomness* di A. Shen, V.A. Uspenski, N. Vereshchagin. Noi ci limiteremo a richiamare quel che serve per lo sviluppo successivo della teoria. Ricordiamo in primo luogo la definizione di semimisura continua

Definizione 1.4.1. Una funzione $f : 2^{<\omega} \to [0,1]$ tale che:

$$f(\sigma) \geq f(\sigma 0) + f(\sigma 1)$$

si dice *semimisura continua.*

Definizione 1.4.2. Una funzione si dice *(inferiormente) semicomputabile*, se vale una delle tre condizioni equivalenti che seguono:
(I) I valori $f(\sigma)$ sono reali inferiormente semicomputabili in modo uniforme rispetto a σ, cioé esiste una funzione computabile $q : 2^{<\omega} \times \mathbb{N} \to \mathbb{Q}_2$ tale che $q(\sigma, i) \leq q(\sigma, i+1)$ e $lim_{i \to +\infty} q(\sigma, i) = f(\sigma)$
(II) Il sottografico di f:

$$\{\langle q, \sigma \rangle \ : \ q \in \mathbb{Q}_2 \wedge \sigma \in 2^{\leq \mathbb{N}} \wedge q < f(\sigma)\}$$

è un insieme c.e.
(III) Esistono funzioni computabili $f^t : 2^{<\omega} \to \mathbb{Q}_2$, $t \in \mathbb{N}$ tali che $f^t(\sigma) \leq f^{t+1}(\sigma)$ e $\lim_{t \to \infty} f^t(\sigma) = f(\sigma)$

Osservazione 1.4.3. Vedremo ora che esiste un'enumerazione effettiva $m_1, m_2, \ldots$ delle semimisure semicomputabili. Pertanto:

$$m(\sigma) = \sum_{i=1}^{\infty} 2^{-i} m_i(\sigma) \tag{1.5}$$

è una semimisura semicomputabile ottimale, nel senso che, data ogni altra semimisura m' esiste una costante c tale che $cm(\sigma) \geq m'(\sigma)$. Sia $m' = m_n$; allora basta prendere $c = 2^n$.

Osservazione 1.4.4. Mostramo che dato un intervallo reale $[a, b]$ esiste un'enumerazione effettiva di tutte le funzioni semicomputabili a valori nel'intervallo. Data una enumerazione W_e di tutti i sottoinsiemi c.e. di $2^{<\omega} \times \mathbb{Q}_2$, poniamo:

$$f_e(\sigma) = \begin{cases} \sup\{r \ : \ (\sigma, r) \in W_e\} & \text{se } a \leq \sup\{r \ : \ (\sigma, r) \in W_e\} \leq b \\ a & \text{se } \sup\{r \ : \ (\sigma, r) \in W_e\} < a \\ b & \text{se } \sup\{r \ : \ (\sigma, r) \in W_e\} > b \end{cases} \tag{1.6}$$

La funzione f_e è inferiormente semicomputabile perché si approssima dal basso colle funzioni computabili:

$$f_{e,s}(\sigma) = \begin{cases} \max\{r : (\sigma, r) \in W_{e,s}\} & \text{se } a \leq \max\{r : (\sigma, r) \in W_{e,s}\} \leq b \\ a & \text{se } \max\{r : (\sigma, r) \in W_{e,s}\} < a \\ b & \text{se } \max\{r : (\sigma, r) \in W_{e,s}\} > b \end{cases}$$

Usiamo la convenzione che il max di un insieme vuoto è $-\infty$. D'altra parte, se f è semicomputabile, il suo sottografico è c.e, per cui f compare nell'enumerazione appena descritta.

Teorema 1.4.5. *Esiste un'enumerazione effettiva delle semimisure inferiormente semicomputabili.*

Dimostrazione. L'idea è quella di prendere un'enumerazione $\{f_e\}_{e\in\mathbb{N}}$ delle funzioni semicomputabili a valori in $[0,1]$ e modificarne le approssimazioni computabili $f_{e,s}$ in modo che soddisfino le condizioni di semimisura. Per ogni s, definiamo l'insieme $Q_s = \{k/2^s : k \in \mathbb{N}\}$. Applichiamo il seguente algoritmo alle approssimazioni $f_{e,s}$:

1. $g_{e,s}(\oslash) = f_{e,s}(\oslash)$
2. $k = 1$
3. Per tutte le stringhe σ di lunghezza k, calcoliamo $f_{e,s}(\sigma)$
4. Per ogni stringa $\tau 0$ di lunghezza k, poniamo:
$$g_{e,s}(\tau 0) = \max\{q \in Q_s : q \leq f_{e,s}(\tau) \wedge q \leq g_{e,s}(\tau)\}$$
5. Per ogni stringa $\tau 1$ di lunghezza k, poniamo:
$$g_{e,s}(\tau 1) = \max\{q \in Q_s : q \leq f_{e,s}(\tau 1) \wedge q + g_{e,s}(\tau 0) \leq g_{e,s}(\tau)\}$$
6. $k = k + 1$ e torniamo a 3

Le funzioni $g_{e,s}$ sono chiaramente computabili; i passaggi 4 e 5 garantiscono che si tratti di semimisure. Inoltre, le $g_{e,s}$ sono crescenti in s; poniamo $g_e(\sigma) = sup_s g_{e,s}(\sigma)$. Si ha che g_e è ancora una semimisura perché le diseguaglianze $g_{e,s}(\sigma) \geq g_{e,s}(\sigma 0) + g_{e,s}(\sigma 1)$ passano al limite. Inoltre, se f_e era una semimisura, $f_e = g_e$. Infatti, supponiamo che $f_e(\tau) = g_e(\tau)$. Allora $\sup_s g_{e,s}(\tau 0) = \sup\{q \in \mathbb{Q}_2 : q \leq f(\tau) \wedge q \leq f(\tau 0)\} = f(\tau 0)$ e analogamente per $g_{e,s}(\tau 1)$. $\square$

Definizione 1.4.6. Sia m una semimisura ottimale; allora si definisce la *complessità a priori* di una stringa σ come:

$$KA(\sigma) = -\log_2(m(\sigma)) \tag{1.7}$$

Osserviamo che se m e $\hat{m}$ sono due semimisure ottimali, differiscono per una costante, per cui, dette KA e $\widehat{KA}$ le complessità a priori che ne derivano, si ottiene:

$$\widehat{KA}(\sigma) = KA(\sigma) \pm O(1)$$

Il significato computazionale da attribuire alle semimisure non è chiaro; verrà ora mostrato che sono la misura immagine della misura di Lebesgue sotto l'azione di un funzionale di Turing. In altre parole, data una semimisura m esiste una funzionale di Turing Φ tale che:

$$m(\sigma) = \lambda(\{X : \Phi(X) \in [\![\sigma]\!]\})$$

Per vederlo, diamo, in primo luogo, le definizioni seguenti:

Definizione 1.4.7. Un insieme di stringhe S si dice albero binario se $\tau \in S$ e $\sigma \preceq \tau$ implicano $\sigma \in S$.
Con $[S]$ s'indica il sottoinsieme di $2^{\mathbb{N}}$ formato dagli elementi X tali che, per ogni n, $X|_n \in S$

Definizione 1.4.8. Siano $S, T \subseteq 2^{<\omega}$ alberi binari. Una funzione $f : S \mapsto T$ si dice monotona se:

$$\sigma, \tau \in S,\ \sigma \preceq \tau \ \Rightarrow\ f(\sigma) \preceq f(\tau)$$

Definizione 1.4.9. Data una funzione monotona f, si può definire la sua estensione $\hat{f} : D(f) \to [T]$, dove

$$D(f) = \{X \in [S] : \lim_{n\to\infty} |f(X|n)| = \infty\}$$

$$\hat{f}(X) = \sup_{\preceq}\{\tau \in S : \exists\sigma \preceq X[f(\sigma = \tau]\}$$

Osservazione 1.4.10. La funzione $\hat{f}$ è continua rispetto alla topologia indotta su $D(f)$ e $[T]$ dallo spazio di Cantor; infatti, sia $\hat{f}(X) = Y$. Sia $y \preceq Y$; Consideriamo i $\tau \succeq y$ tali che $\exists\sigma \preceq X\ [f(\sigma) = \tau]$ e fra essi scegliamo quello di minima lunghezza, che chiameremo τ'. Sia $\sigma' \preceq X$ tale che $f(\sigma') = \tau'$; allora $\hat{f}([\![\sigma']\!]) \subseteq [\![\tau']\!]$

Data una funzione monotona f, possiamo definire il suo *sottografico*, cioè l'insieme Γ delle coppie (σ, τ) tali che $\tau \preceq f(\sigma)$.

Definizione 1.4.11. Se l'insieme Γ è c.e, la funzione si dice *ricorsiva.* In tal caso, se ne definiscono approssimazioni computabili come segue:

$$f_s(\sigma) = \max_{\preceq}\{\tau\ : (\sigma, \tau) \in \Gamma_s\}$$

dove il massimo è posto uguale a $\oslash$ se l'insieme è vuoto.

Osservazione 1.4.12. Una funzionale di Turing Φ induce una funzione monotona $\phi : 2^{<\omega} \to 2^{<\omega}$. Sia $\sigma \in 2^{<\omega}$. Consideriamo l'insieme

$$A_\sigma = \{y \in 2^{<\omega}\ :\ \exists x \in 2^{<\omega}\ ,\ x \succeq \sigma[\Phi^x \downarrow\succeq y]\}$$

Sia τ la più lunga stringa che è prefisso comune a tutti gli elementi di A_σ. Poniamo $\phi(\sigma) = \tau$. Non è difficile vedere che, nelle notazioni precedenti, $\hat{\phi} = \Phi$.
Infatti, sia $\Phi^X = Y$ e $y \preceq Y$. Allora esiste un $\sigma \preceq X$ tale che $\Phi^\sigma \downarrow\succeq y$ e, per ogni estensione x di σ $\Phi^x \succeq y$ (ciò deriva dal cosiddetto "principio dell'uso"). Allora sicuramente y è prefisso comune di tutti gli elementi di A_σ, per cui $\phi(\sigma) \succeq y$, dal che si deduce che $\hat{\phi}(X) \succeq y$.

Osservazione 1.4.13. Se Φ è un funzionale di Turing, la ϕ monotona associata è ricorsiva. Infatti, per enumerare il suo sottografico Γ si può procedere come segue:

$$\Gamma_s = \{(\sigma, \tau); |\sigma|, |\tau| \leq s, \Phi_s^\sigma \downarrow\succeq \tau\}$$

Viceversa, data una funzione monotona ricorsiva, essa si trova come ϕ associata a un qualche funzionale di Turing Φ. Basta prendere come Phi il funzionale che, con oracolo X, enumera il grafico Γ e vi cerca le coppie (x, y) con $x \preceq X$; quando ne trova una più lunga dell'output attuale, ne scrive le cifre successive a quelle dell'output corrente.
In definitiva, si possono identificare le funzioni monotone ricorsive con i funzionali di Turing.

Data una funzione monotona f, definiamo la funzione $\lambda_f : 2^{<\omega} \to \mathbb{R}^{\geq 0}$:

$$\lambda_f(x) = \lambda([\![\{\sigma\ :\ \exists\tau \succeq x[f(\sigma) \succeq \tau]\}]\!])$$

Abbiamo il seguente:

Teorema 1.4.14. 1. *Se f è una funzione monotona ricorsiva, λ_f è una semimisura inferiormente semicomputabile.*

2. *Se m è una semimisura inferiormente semicomputabile, esiste una funzione monotona ricorsiva f tale che $m = \lambda_f$.*

Cominciamo dal punto 1.

Abbiamo:

$$\lambda_f(\oslash) = \lambda([\![\{\sigma \,:\, \exists\tau[f(\sigma) \succeq \tau]\}]\!]) \leq 1$$

Inoltre,

$$\begin{aligned}&[\![\{\sigma \,:\, \exists\tau \succeq x[f(\sigma) \succeq \tau]\})]\!] \supseteq \\ &[\![\{\sigma \,:\, \exists\tau \succeq x0[f(\sigma) \succeq \tau]\}]\!] \cup [\![\{\sigma \,:\, \exists\tau \succeq x1[f(\sigma) \succeq \tau]\}]\!]\end{aligned}$$

da cui, siccome la monotonia di f implica che $[\![\{\sigma \,:\, \exists\tau \succeq x0[f(\sigma) \succeq \tau]\}]\!]$ e $[\![\{\sigma \,:\, \exists\tau \succeq x1[f(\sigma) \succeq \tau]\}]\!]$ sono disgiunti:

$$\lambda_f(x) \geq \lambda_f(x0) + \lambda_f(x1).$$

Infine, per vedere che la semimisura è semicomputabile, basta osservare che è approssimata dal basso delle funzioni computabili:

$$\lambda_{f,s} = \lambda([\![\{\sigma \,:\, \exists\tau \succeq x[f_s(\sigma) \succeq \tau]\})]\!])$$

Procediamo alla dimostrazione di 2, per la quale sarà necessaria una serie di lemmi.

Lemma 1.4.15. *Sia m una semimisura continua inferiormente semicomputabile. Esiste allora una famiglia $\{m_i\}_{i\in\mathbb{N}}$ di approssimazioni computabili tali che:*

1.

$$m_i : 2^{<\omega} \to \mathbb{Q}_2$$

2.

$$\lim_{i\to\infty} m_i(\sigma) = m(\sigma)$$

3.

$$m_i(\sigma) \leq m_{i+1}(\sigma)$$

4. Per ogni i, la funzione m_i è una semimisura continua che vale 0 su tutte le stringhe di lunghezza maggiore di i.

Dimostrazione. Segue dalla definizione di funzione semicomputabile che esistono a_i che soddisfano 1,2, e 3. A meno si sostituire con 0 ogni valore negativo, si può supporre che tutte le approssimazioni siano non negative. Inoltre, a meno di moltiplicare a_i per $1 - 2^| - i|$ si può supporre che $a_i(\sigma) < m(\sigma)$. Modificheremo le funzioni a_i, che già soddisfano i requisiti 1, 2 e 3, in modo che soddisfino anche 4. Definiamo m_i ricorsivamente.

1. $m_i(\sigma) = 0$ se $|\sigma| > i$
2. Sia $k = i$
3. Abbiamo già determinato $m_i(\sigma) \in \mathbb{Q}_2$ per $|\sigma| > k$. Sia σ di lunghezza k. Allora:
4. Se $a_i(\sigma) \geq m_i(\sigma 0) + m_i(\sigma 1)$, poniamo $m_i(\sigma) = a_i(\sigma)$. Altrimenti, poniamo $m_i(\sigma) = m_i(\sigma 0) + m_i(\sigma 1)$
5. Poniamo $k = k - +1$ e torniamo al passo 3

□

Diremo che una semimisura è *semplice* se è a valori nei razionali diadici ed è non nulla solo su un insieme finito di stringhe. Useremo, nel resto del paragrafo, l'identificazione abituale della stringa σ con l'intervallo diadico, detto *elementare*, $I_\sigma = [\sum \sigma(i)2^{-i}, \sum \sigma(i)2^{-i} + 2^{-|\sigma|}[$. Diremo che un sottoinsieme di $[0,1]$ è *semplice* se è l'unione di un numero finito di intervalli del tipo I_σ. Infine, una *famiglia semplice* sarà una famiglia $\{A_\tau\}$ di sottoinsiemi semplici di $[0,1]$, tali che solo un numero finito di essi è non vuoto e $A_\tau \supseteq A_{\tau 0} \cup A_{\tau 1}$, $A_{\tau 0} \cap A_{\tau 1} = \emptyset$.

Lemma 1.4.16. *Data una semimisura semplice s, esiste una famiglia semplice tale che:*

$$s(\sigma) = \lambda(A_\sigma)$$

Tale famiglia si trova da s con un procedimento effettivo.

Dimostrazione. Usiamo il seguente algoritmo:

- Prendiamo come $A_\oslash$ un intervallo di lunghezza $s(\oslash)$ che sia un sottoinsieme semplice di $[0,1]$, il che è possibile perché $s(\oslash)$ è un razionale diadico.
- Supponiamo d'aver definito A_σ. Esso è unione finita di intervalli diadici elementari e disgiunti la cui misura totale è $s(\sigma)$; siccome $s(\sigma 0) + s(\sigma 1) \leq s(\sigma)$, è possibile trovare dentro A_σ due famiglie finite e disgiunte di intervalli diadici elementari e disgiunti tale che la prima ha misura $s(\sigma 0)$ e la seconda $s(\sigma 1)$. Assegnamo alla prima nome $A_{\sigma 0}$ e alla seconda $A_{\sigma 1}$

Vediamo meglio come viene eseguito il secondo passaggio. Sappiamo che $s(\sigma), s(\sigma 0), s(\sigma 1)$ si possono scrivere, rispettivamente, come $n/2^k$, $r/2^k$ e $s/2^k$ con $r + s \leq n$. Suddividiamo A_σ in intervalli di lunghezza 2^{-k}. Assegneremo a $A_{\sigma 0}$ i primi r intervalli e a $A_{\sigma 1}$ i successivi s. La procedura è chiaramente effettiva. □

Lemma 1.4.17. *Siano s_1, s_2 semimisure semplici. Sia $\{B_\sigma\}$ la famiglia semplice associata a s_1. Se $s_1 \leq s_2$, è possibile costruire una famiglia semplice $\{C_\sigma\}$ tale che $B_\sigma \subseteq C_\sigma$ e $s_2(\sigma) = \lambda(C_\sigma)$.*

Dimostrazione. Usiamo il seguente algoritmo:

- Prendiamo come $C(\oslash)$ un intervallo di lunghezza $s_2(\oslash)$ che sia un insieme semplice e che contenga $B_\oslash$ (ciò è possibile perché $s_2(\oslash) \geq s_1(\oslash)$).

- Supponiamo d'aver definito C_σ. Esso è unione finita di intervalli diadici elementari disgiunti la cui misura totale è $s_2(\sigma)$ e contiene B_σ. A sua volta, B_σ contiene insiemi semplici disgiunti $B_{\sigma 0}, B_{\sigma 1}$. Poiché $s_1(\sigma 0) \leq s_2(\sigma 0)$, $s_1(\sigma 1) \leq s_2(\sigma 1)$ e $s_2(\sigma 0) + s_2(\sigma 1) \leq s(\sigma)$, è possibile trovare dentro C_σ due famiglie finite e disgiunte di intervalli diadici elementari e disgiunti tale che la prima ha misura $s(\sigma 0)$ e contiene $B_{\sigma 0}$, mentre la seconda ha misura $s(\sigma 1)$ ed contiene $B_{\sigma 1}$. Assegnamo alla prima nome $C_{\sigma 0}$ e alla seconda $C_{\sigma 1}$.

□

Lemma 1.4.18. *E' possibile trovare una famiglia di insiemi semplci $\{A_{\sigma,i}\}$ in modo che:*

1. *$A_{\sigma,i}$ è computabile da σ e i*
2. *$\lambda(A_{\sigma,i}) = m_i(\sigma)$*
3. *$A_{\sigma 0,i} \cap A_{\sigma 1,i} = \emptyset$*
4. *$A_{\sigma,i} \subseteq A_{\sigma,i+1}$*

Dimostrazione. Basta applicare ripetutamente il lemma precedente alle semimisure computabili e semplici m_i □

Lemma 1.4.19. *Se m è una semimisura inferiormente semicomputabile, esiste una funzione monotona ricorsiva f tale che $m = \lambda_f$.*

Dimostrazione. Descriveremo ora un algoritmo effettivo Φ che definisce la funzione monotona ricorsiva f tale che $m = \lambda_f$. Φ agisce come segue sulla stringa σ:

1. Costruisce tutti gli insiemi $A_{\tau,i}$ per $|\tau| \leq |\sigma|$ e $i \leq |\sigma|$.

2. Cerca tutte le stringhe x tali che $\sigma \in A_{x,i}$ (identificando σ con $\sum \sigma(i)2^{-i}$. Osserviamo che tali stringhe sono compatibili perché $A_{x,i} \cap A_{y,j} \neq \emptyset$ implica che x e y sono compatibili (cioé una delle due è prefisso dell'altra).

3. $\Phi(\sigma)$ sarà allora la più lunga fra le stringhe trovate al passo precedente.

La stringa $f(\sigma)$ ha per prefisso con la stringa z se e solo σ sta in $\cup_i A_{z,i}$, per cui:

$$\{\sigma \ : \ \exists\tau \succeq x[f(\sigma) \succeq \tau]\} = \{\sigma \ : \sigma \in \ \cup_i A_{z,i}\}$$

Ora, l'insieme $\cup_i A_{z,i}$ è unione di intervalli del tipo I_ρ, per cui si può trovare un' *anticatena* R (cioé un insieme di stringhe a due a due incompatibili) tale che:

$$\cup_i A_{z,i} = \cup_{\rho \in R} I_\rho$$

Ciò implica che:

$$[\![\{\sigma \ : \sigma \in \ \cup_i A_{z,i}\}]\!] = [\![R]\!]$$

Allora :

$$\lambda([\![\{\sigma \ : \ \exists\tau \succeq x[f(\sigma) \succeq \tau]\}]\!]) = \lambda([\![R]\!]) = \sum_{\rho \in R} 2^{-|\rho|} = \lambda(\cup_i A_{z,i}) = m(z)$$

□

Osservazione 1.4.20. Il significato intutitivo da assegnare alle semimisure continue è ora chiaro. La semimisura associata a un certo funzionale Φ_e associa a σ la probabilità che, scegliendo l'input X con distribuzione uniforme, l' output cominci con σ. Il teorema precedente dice che tutte le semimisure nascono in questo modo.

Capitolo 2

Casualità algoritmica

2.1 Il concetto di casualità algoritmica

In questo paragrafo definiremo il concetto di casualità algoritmica secondo Martin-Löf (o 1-randomness). Sono possibili diversi approcci, che portano a caratterizzazioni fra loro equivalenti. La prima definizione fa uso della misura di Lebesgue sullo spazio di Cantor $2^{\mathbb{N}}$. Ricordiamo che tale misura viene indicata col simbolo λ e, talora, identificata con la corrispondente premisura sulle stringhe di $2^{<\omega}$. Richimiamo, inoltre, alcune nozioni sugli insiemi e le classi Σ_1^0. Data una famiglia $\{U_e\}$ di insiemi c.e, essa si dice *uniformemente c.e* se esiste una funzione computabile totale f tale che $U_e = W_{f(e)}$ (ricordiamo, a tal proposito, che $\{W_e\}_{e \in \mathbb{N}}$ è l'enumerazione canonica degli insiemi c.e definita nel parafrafo 1.3). Ciò significa, equivalentemente, che esiste una funzione computabile totale $g(e, n)$, che, fissato l'argomento e, al variare di n enumera tutti gli elementi di U_e.
Una classe A dello spazio di Cantor è detta Σ_1^0 se esiste un insieme c.e. W_e tale che $A = [\![W_e]\!]$. Ciò è equivalente a richiedere che esista una relazione computabile R tale che:

$$X \in A \Leftrightarrow \exists y R(X|_y).$$

Osserviamo che le classi Σ_1^0 sono effettivamente aperte nel senso che esiste una macchina di Turing Φ_e che enumera gli intorni della base di $2^{\mathbb{N}}$ la cui unione forma la classe data. Il complementare di una classe Σ_1^0 è detto classe Π_1^0; si tratta, dunque, di un insieme effettivamente chiuso.

Definizione 2.1.1. Sia $\{R_n\}$ una successione di insiemi uniformemente c.e. La successione di classi Σ_1^0 data da $G_n = [\![R_n]\!]$ viene detta *successione di classi uniformemente* Σ_1^0.

Definizione 2.1.2. Una successione $\{G_n\}$ di classi uniformemente Σ^0_1 si dice essere un *test di Martin-Löf* (in breve, un test ML), se per ogni n si ha:

$$\lambda(G_n) \leq 2^{-n}$$

Si dice che $Z \in 2^{\mathbb{N}}$ *fallisce* nel test se $Z \in \bigcap_n G_n$; in caso contrario, si dice che Z *supera* il test.
Si dice che Z è *casuale nel senso di Martin-Löf* se supera tutti i test ML. La classe di tutti gli insiemi ML si indica con MLR e il suo complementare con non-MLR.

Il significato intuitivo di questa definizione è chiaro: un test rappresenta un tentativo di individuare Z (a meno di un insieme molto "piccolo", cioè di misura nulla in senso effettivo) e Z viene detto casuale se sfugge a tutti questi tentativi.
Ricordiamo (si veda, ad esempio, [5], pagg.54-55) che, data una classe Σ^0_1 $A = [\![W_e]\!]$, se ne può definire un'approssimazione effettiva ponendo:

$$A_s = \{x : \quad |x| \leq s \wedge \exists y \in W_{e,s}[y \preceq x]\}.$$

Allora si vede che $A = \bigcup_s [\![A_s]\!]$. Questo induce, d'altra parte, un'approssimazione effettiva dall'alto della classe Π^0_1 data da $P = 2^{\mathbb{N}} \backslash A$. Tale approssimazione è individuata dagli insiemi:

$$P_s = [\![\{x : \quad |x| = s \wedge x \notin W_{e,s}\}]\!]$$

Si ha che $P_s \supseteq P_{s+1}$ e $P = \bigcap_s P_s$.

Esempio 2.1.3. Se Z è computabile, $G_n = [\![Z|_n]\!]$ è un test ML che Z non supera, per cui gli insiemi computabili non sono MLR.
Più in generale, sia P una classe Π^1_0 di misura nulla. Considerando la sua approssimazione P_s prima definita, si ha che $\lim_s \lambda(P_s) = \lambda P = 0$, per cui si può trovare una funzione computabile f tale che $\lambda(P_{f(n)}) \leq 2^{-n}$. Allora $G_n = P_{f(n)}$ è un test ML. Pertanto,ogni membro di una classe Π^1_0 di misura nulla. è non-MLR.

Definizione 2.1.4. Un test ML $\{U_n\}$ si dice *universale* se, per ogni altro test ML $\{G_n\}$, $\bigcap_n G_n$ è contenuto in $\bigcap_n U_n$.

In altre parole, MLR=$2^{\mathbb{N}} - \bigcap_n U_n$.

Teorema 2.1.5. *Esiste un test ML universale.*

Dimostrazione. Sia $\{R_n^0\}, \{R_n^1\}, \ldots$ un'enumerazione effettiva delle successioni di insiemi uniformemente c.e. Enumeriamo gli elementi di R_n^i nell'insieme S_n^i finché viene rispettata la condizione $\lambda(\llbracket S_n^i \rrbracket) \leq 2^{-n}$. In altre parole, definiamo una successione $\{S_n^i\}$:

$$\begin{cases} S_{n,s}^i = R_{n,s}^i & \text{se } \lambda(\llbracket R_{n,s}^i \rrbracket) \leq 2^{-n} \\ S_{n,s}^i = S_{n,s-1}^i & \text{altrimenti} \end{cases}$$

Allora, $L_n^i = \llbracket S_n^i \rrbracket$ è una enumerazione effettiva di tutti i test ML. Poniamo $U_n = \bigcup_i L_{n+i+1}^i$. Si vede che $X \in U_n$ se e solo se $\exists i \exists s\ [X \in \llbracket L_{n+i+1,s}^i \rrbracket]$, per cui $\{U_n\}$ è una successione di classi uniformemente Σ_1^0. Inoltre:

$$\lambda(U_n) \leq \sum_i \lambda(L_{n+i+1}^i) \leq \sum_i 2^{-n-i-1} \leq 2^n$$

Pertanto, $\{U_n\}$ è un test ML. Infine, sia $Z \in \bigcap_n G_n$. C'è un j tale che $G_n = L_n^j$. Abbiamo che, per ogni n, $G_{n+j+1} \subseteq U_n$, per cui $Z \in U_n$. □

È possibile dare una nozione di casualità imponendo che i segmenti iniziali della stringa Z siano incomprimibili, cioè che esista un b tale che:

$$\forall n\ [K(Z|_n) > n - b]$$

dove K è la complessità di Kolmogorov rispetto a una macchina senza prefissi universale.
Questa condizione è equivalente a quella di superare tutti i test di Martin-Löf (o, equivalentemente, un test ML universale).

Teorema 2.1.6. *Le affermazioni seguenti sono equivalenti:*
(I) Z è MLR
(II) $\forall n\ [K(Z|_n) > n - b]$

Per vederlo avremo bisogno di un lemma:

Lemma 2.1.7. *Poniamo, per $b \in \mathbb{N}$, $R_b = \llbracket \{x \in 2^{<\omega} : K(x) \leq |x| - b\} \rrbracket$. Allora $\{R_b\}_{b \in \mathbb{N}}$ è un test ML.*

Dimostrazione. Fissiamo una certa macchina universale senza prefissi U, in modo che $K(x) = K_U(x)$. Osserviamo che la condizione $K_U(x) \leq |x| - b$ significa:

$$\exists \sigma \exists s [U_s(\sigma) = x \wedge |\sigma| \leq |x| - b]$$

La proprietà quantificata esistenzialmente è effettiva, perché per verificare se sussista basta eseguire U su input σ per s passi e, se si arresta, confrontare il suo output con x. Pertanto, la condizione su x è Σ^0_1 e, dipendendo essa dal parametro b, $\{R_b\}$ è una successione di classi uniformemente Σ^0_1. Sia $S_b = \{x \in 2^{<\omega} : K(x) \leq |x| - b\}]\!]$; per ogni $x \in S_b$ esiste allora un x^* con $|x^*| \leq |x| - b$ e $U(x^*) = x$, per cui:

$$\lambda(R_b) \leq \sum_{x \in S_b} 2^{-|x|} \leq \sum_{U(x^*)\downarrow} 2^{-|x^*|-b} \leq 2^{-b}$$

Nell'ultimo passaggio abbiamo usato il fatto che il dominio di U è senza prefissi, per cui:

$$\sum_{U(\sigma)\downarrow} 2^{-|\sigma|} = \lambda(dom(U)) \leq 1$$

□

Dimostrazione. (del teorema 2.1.6)
Si vede subito che (I) implica (II), perché,se Z è MLR, deve superare il test ML definito nel lemma precedente, per cui esiste un b per cui $Z \notin R_b$.
Vediamo ora che non (I) implica non (II). A partire da un test ML $\{G_n\}$ che Z non supera, costruiremo un insieme accettabile di condizioni per una macchina senza prefissi M in modo che $K_M(Z|_n) \leq n - O(1)$. A meno di sostituire n con $2n$, possiamo supporre che $\lambda(G_n) \leq 2^{-2n}$. Per ogni G_n si può ottenere in modo effettivo un'anticatena C_n in modo che $G_n = [\![C_n]\!]$. Ora per ogni elemento $x^n_i \in C_n$ enumeriamo una condizione $\langle |x^n_i| - n + 1, x^n_i \rangle$; sia L l'insieme di tutte queste condizioni. Osserviamo che la definizione è ben posta, perché tutte le stringhe di C_n devono avere lunghezza almeno $2n$, affinché $\lambda(G_n) \leq 2^{-2n}$. Abbiamo che:

$$\sum_{x^n_i \in \cup C_n} 2^{-(|x^n_i|-n+1)} = \sum_n \sum_{x^n_i \in C_n} 2^{-(|x^n_i|-n+1)} = \sum_n 2^{n-1}\lambda(G_n) \leq 1$$

Pertanto, L è un insieme accettabile di richieste (nell'accezione del teorema di Kolmogorov-Chaitin, per cui si veda, ad esempio, il capitolo 3 di [6] e si può costruire una macchina senza prefissi M tale che, per ogni x^n_i, esiste σ con $|\sigma| \leq |x^n_i| - n + 1$ tale che $M(\sigma) = x^n_i$. Sappiamo che esiste una d tale che $K(x) \leq K_M(x) + d$. Sia ora $b \in \mathbb{N}$. Poniamo $m = b + d + 1$. Si ha che $Z \in G_m$, per cui esiste un n e un $x^m_i \in C_m$ tale che $Z|n = x^m_i$. Si ha, pertanto:

$$K(Z|n) \leq K_M(x^m_i) + d \leq |x^m_i| - b - d - 1 + 1 + d = n - b$$

□

Osservazione 2.1.8. Il teorema precedente implica che il test $\{R_b\}$ è universale.

Un altro approccio alla definizione di casualità deriva dall'idea che una stringa rappresenti la sequenza di "testa" o "croce" che viene prodotta dal lancio ripetuto di una moneta. Su ciascun lancio è possibile scommettere seguendo una certa strategia, rappresentata da una funzione che associa a ogni stringa il capitale in possesso del giocatore se si verificano quei dati esiti. Una strategia equa, nel senso che il valore atteso del capitale del giocatore dopo il prossimo lancio è pari al capitale di cui è attualmente in possesso, è rappresentata da una funzione detta martingala. Se la stringa è imprevedibile, deve essere impossibile, qualunque strategia equa o sfavorevole si adotti, conseguire vincite arbitrariamente elevate.

Definizione 2.1.9. Una funzione $f : 2^{\mathbb{N}} \mapsto \mathbb{R}^{\geq 0} \cup \{+\infty\}$ si dice *martingala* se vale la condizione:

$$f(\sigma) = \frac{f(\sigma 0) + f(\sigma 1)}{2}$$

Si dice *supermartingala* se

$$f(\sigma) \geq \frac{f(\sigma 0) + f(\sigma 1)}{2}$$

Definizione 2.1.10. Una supermartingala si dice semicomputabile inferiormente (o c.e.) se è una funzione inferiormente semicomputabile.

Abbiamo i risultati seguenti:

Lemma 2.1.11 (Disuguaglianze di Kolmogorov)**.** *Sia d una (super)martingala. (I) Sia data un'anticatena C di estensioni di una certa stringa σ; vale allora:*

$$\sum_{\tau \in C} 2^{-|\tau|} d(\tau) \leq 2^{-|\sigma|} d(\sigma)$$

(II) Sia $R_k = \{\sigma \ : \ d(\sigma) \geq k\}$. Allora $\lambda([\![R_k]\!]) \leq \frac{d(\emptyset)}{k}$

Dimostrazione. Per dimostrare (I) basta considerare C finita; infatti, se C fosse infinita, basterebbe considerare la successione crescente per inclusione di anticatene finite $C_n = 2^{\leq n} \cap C$ e passare al limite. Procediamo per induzione sulla cardinalità di C. Se $|C| = 1$, il suo unico elemento τ è un'estensione di σ: $\tau = \widehat{\sigma\beta}$ con $|\beta| = |\tau| - |\sigma|$. Applicando ripetutamente il fatto che $2d(\sigma) \geq d(\sigma i)$, $i = 0, 1$, si ottiene che $d(\tau) = d(\sigma\hat{\ }\beta) \leq 2^{|\beta|} d(\sigma) = 2^{|\tau|-|\sigma|} d(\sigma)$. Supponiamo, ora, che (I) valga per $|C| \leq n$. Sia $|C| = n + 1$. Esiste una stringa di massima lunghezza $\nu \succeq \sigma$ tale che tutti gli elementi di C sono sua estensione. Poniamo $C_i = \{\tau \in C \ : \ \tau \succeq \nu i\}$, $i = 0, 1$. Per la massimalità di ν, $C_i \neq \emptyset$, il che implica $|C_i| \leq n$, per cui, usando l'ipotesi induttiva:

$$\sum_{\tau \in C} 2^{-|\tau|} d(\tau) = \sum_{\tau \in C_0} 2^{-|\tau|} d(\tau) + \sum_{\tau \in C_1} 2^{-|\tau|} d(\tau) \leq 2^{-|\nu|-1} d(\nu 0) + 2^{-|\nu|-1} d(\nu 1)$$
$$\leq 2^{-|\nu|} d(\nu) \leq 2^{-|\sigma|} d(\sigma)$$

dove la prima disuguaglianza segue dall'ipotesi induttiva e l'ultima da quanto già provato per il caso $|C| = 1$.

Per il punto (II), basta prendere un'anticatena $C_k \subseteq R_k$ che generi $[\![R_k]\!]$; allora, dal punto (I) applicato a $\sigma = \oslash$, si ottiene:

$$\lambda([\![C_k]\!]) = \sum_{\tau \in C_k} 2^{-|\tau|} \frac{d(\tau)}{d(\tau)} \leq \sum_{\tau \in C_k} 2^{-|\tau|} \frac{d(\tau)}{k} \leq \frac{d(\oslash)}{k}$$

□

Osservazione 2.1.12. Le martingale sono strettamente associate con le misure di probabilità. Infatti, se d è una martingala, si vede subito dalla definizione che $2^{|\sigma|} d(\sigma)$ è la premisura associata a una misura di probabilità.

Esempio 2.1.13. Sia B una classe misurabile. Poniamo:

$$d_B(\sigma) = 2^{|\sigma|} \lambda([\![\sigma]\!] \cap B) = \lambda(B|\sigma), \tag{2.1}$$

dove $\lambda(B|\sigma)$ indica la probabilità di B condizionata a $[\![\sigma]\!]$. Dall'additività della misura di Lebesgue segue che:

$$d_B(\sigma) = 2^{|\sigma|} \lambda([\![\sigma]\!] \cap B) = 2^{|\sigma|} \lambda([\![\sigma 0]\!] \cap B) + 2^{|\sigma|} \lambda([\![\sigma 1]\!] \cap B) = \frac{d_B(\sigma 0) + d_B(\sigma 1)}{2}$$

per cui d_B è una martingala. Se $B = [\![W_e]\!]$ è una classe Σ^0_1, la martingala associata è inferiormente semicomputabile. Infatti, sia $B_s = [\![W_{e,s}]\!]$ un'approssimazione di B. Poiché $W_{e,s}$ è finito, è possibile determinare effettivamente un'anticatena finita C_s tale che $[\![\sigma]\!] \cap B_s = [\![C_s]\!]$, cosicché la funzione:

$$d^s_B(\sigma) = 2^{|\sigma|} \lambda([\![\sigma]\!] \cap B_s) = 2^{|\sigma|} \sum_{\tau \in C_s} 2^{-|\tau|}$$

è computabile. Inoltre, la continuità dal basso della misura di Lebesgue, assieme al fatto che $([\![\sigma]\!] \cap B) = \bigcup_s ([\![\sigma]\!] \cap B_s)$, con $[\![\sigma]\!] \cap B_s \subseteq [\![\sigma]\!] \cap B_{s+1}$, permette di concludere che $lim_{s \to \infty} d^s_B(\sigma) = d_B(\sigma)$.

Proposizione 2.1.14. *(I) Se $\alpha \geq 0$, d_1 e d_2 sono (super)martingale, αd_1 e $d_1 + d_2$ sono ancora (super)martingale.*
(II) Se $d_1, d_2, \ldots$ è una successione di (super)martingale tali che $\sum_i d_i(\emptyset) < +\infty$, porre $d(\sigma) = \sum_i d_i(\sigma)$ definisce una (super)martingala. Se le d_i sono uniformemente c.e in i, d è c.e.

Dimostrazione. La (I) è una semplice verifica. Per la (II), si può notare che per ogni σ,

$$\sum_i d_i(\sigma) \leq 2^{|\sigma|} \sum_i d_i(\oslash) < +\infty,$$

per cui la definizione è ben posta; che d sia una (super)martingala viene dalle proprietà di linearità delle serie convergenti. Se le d_i sono uniformemente c.e, ponendo: $d^s(\sigma) = \sum_{1 \leq i \leq s} d_i^s(\sigma)$ si ottengono funzioni computabili che approssimano d dal basso. □

Definizione 2.1.15. Sia d una supermartingala. Si dice che d *ha successo* su $Z \in 2^{\mathbb{N}}$ se

$$\sup_{n \geq 0} d(Z|_n) = +\infty$$

La classe degli Z su cui d ha successo si indica con $S[d]$.

Osservazione 2.1.16. Data una martingala d, per ogni $\alpha \in \mathbb{R}^{>0}$, si ottiene che $S[d] = S[\alpha d]$

Teorema 2.1.17. *Le condizioni seguenti si equivalgono:*
(I) L'insieme Z è MLR
(II) Non esiste alcuna (super)martingala inferiormente semicomputabile d che abbia successo su Z.

Dimostrazione. Dimostriamo, in primo luogo, che (I) implica (II).
Sia $Z \in$ MLR e d (super)martingala c.e. Costruiamo, a partire da d un test ML. Possiamo supporre che $d(\oslash) \leq 1$, perché, in caso contrario si può moltiplicare d per una costante abbastanza piccola senza modificare la classe $S[d]$. Poniamo, per ogni $k \in \mathbb{N}$, $G_k = [\![R_k]\!] = [\![\{\sigma \ : \ d(\sigma) > 2^k\}]\!]$. Dalla seconda disuguaglianza di Kolmogorov segue che $\lambda(G_k) \leq \frac{d(\emptyset)}{2^k} \leq \frac{1}{2^k}$. Inoltre G_k è una successione di classi uniformemente Σ_1^0, poiché, dalle approssimazioni computabili d^s di d, che esistono perché d è supposta inferiormente semicomputabile, si può costruire una enumerazione effettiva e uniforme in k degli insiemi R_k:

$$R_{k,s} = \{\sigma \ : \ |\sigma| \leq s \wedge d^s(\sigma) > 2^k\}$$

Pertanto, $\{G_k\}$ è un test ML. Poiché Z è per ipotesi MLR, esiste un $\bar{k}$ tale che $Z \notin G_{\bar{k}}$, cioè per ogni $n \in \mathbb{N}$, $Z|_n \notin R_{\bar{k}}$, il che significa che $\forall n \in \mathbb{N}\ d(Z|_n) \leq 2^{\bar{k}}$, per cui $sup_n d(Z|_n) < +\infty$ e $Z \notin S[d]$.
Mostriamo che (II) implica (I). Sia dato Z tale che per ogni d (super)martingala inferiormente semicomputabile $Z \notin S[d]$. Prendiamo un test ML $\{G_k\}$. Possiamo definire martingale inferiormente semicomputabili d_{G_k} come nell'esempio 2.1.13. Inoltre, $d_{G_k}(\emptyset) = \lambda(G_k) \leq 2^{-k}$, per cui $\sum_{k\geq 1} d_{G_k}(\emptyset) \leq \sum_k 2^{-k} \leq 1$. Per il punto (II) della proposizione precedente, $d = \sum_k d_{G_k}$ è una martingala e dunque anche una supermartingala ed è anche inferiormente semicomputabile. In particolare, $Z \notin S[d]$. Osserviamo che, se esiste $n \in \mathbb{N}$ tale che $[\![Z|_n]\!] \subseteq G_k$, si ha che $d_k(Z|_n) = 2^n \lambda(G_k \cap [\![Z|_n]\!]) = 2^n \lambda([\![Z|_n]\!]) = 1$. Se si verificasse che per ogni k $Z \in G_k$, per ogni intero $M > 0$ si potrebbe trovare un n tale che $\forall k \leq M\ [[\![Z|_n]\!] \subseteq G_k]$. Questa significa che

$$d(Z|_n) = \sum_{k\geq 0} d_{G_k}(Z|_n) \geq \sum_{0\leq k\leq M} d_{G_k}(Z|_n) \geq M+1$$

Pertanto, $sup_n d(Z|_n) = +\infty$, ma questo è assurdo, per cui deve esistere un k per cui $Z \notin G_k$. □

Osservazione 2.1.18 (Universalità e ottimalità)**.** Abbiamo visto in precedenza che esistono test ML universali. Usando la precedura indicata nella dimostrazione del teorema precedente, da un test ML universale si può costruire una martingala universale f, colla proprietà che data ogni altra martingala d, $S[d] \subseteq S[f]$. Una condizione sufficiente, ma non necessaria affinché una (super)martingala f sia universale è che data un'altra (super)martingala d, esista una costante c tale che $cf(\sigma) \geq d(\sigma)$. Una (super)martingala con questa proprietà è detta *ottimale*; vedremo ora che esiste una supermartingala ottimale, mentre, come mostrato nel teorema 6.3.12 di [6], non esiste alcuna martingala ottimale.

Data una supermartingala d, la funzione $d'(\sigma) = 2^{-|\sigma|}d(\sigma)$ risulta essere una semimisura; viceversa data una semimisura, moltiplicando per $2^{|\sigma|}$ si trova una supermartingala; se una delle due è inferiormente semicomputabile, lo è ovviamente anche l'altra.

Lemma 2.1.19. *Esiste un'enumerazione effettiva di tutte le supermartingale c.e che valgono al più $2^{|\sigma|}$ sulla stringa σ.*

Dimostrazione. Osserviamo che se una supermartingala vale al più $2^{|\sigma|}$ sulla stringa σ, tutti i valori della semimisura associata alla sono maggiorati da 1 e viceversa. Enumerare le supermartingale c.e. con la proprietà cercata equivale, pertanto, ad enumerare le semimisure c.e maggiorate da 1, il che è già stato fatto nel capitolo 1. □

Teorema 2.1.20. *Esiste una supermartingala ottimale.*

Dimostrazione. Sia $d_0, d_1, \ldots$ un'enumerazione effettiva di tutte le supermartingale c.e. che valgono al più $2^{|\sigma|}$ sulla stringa σ. Poniamo:

$$d(\sigma) = \sum_{i=0}^{\infty} 2^{-i} d_i(\sigma)$$

Osserviamo che $\sum_{i=0}^{\infty} 2^{-i} d_i(\emptyset) \leq \sum_{i=0}^{\infty} 2^{-i} < \infty$, per cui, in virtù della proposizione, d è in effetti una supermartingala c.e. Sia f un'altra supermartingala; la supermartingala $f' = \frac{f}{f(\emptyset)}$ vale al più $2^{|\sigma|}$ sulla stringa σ, per cui apparirà nell'enumerazione con un certo indice i; allora:

$$d(\sigma) \geq 2^{-i} d_i(\sigma) = \frac{2^{-i}}{f(\emptyset)} f(\sigma)$$

□

2.2 Classi Π^0_1 di misura positiva

Esamineremo ora quali classi Π^0_1 contengono elementi casuali. Come detto nell'esempio 1.1.3, se una classe Π^0_1 ha misura nulla, nessun suo membro è MLR. D'altra parte, se P (non necessariamente Π^0_1) ha misura positiva, $\lambda(P) > 0$, $\lambda(P \cap \mathrm{MLR}) \geq \lambda(P) + \lambda(\mathrm{MLR}) - \lambda(P \cup \mathrm{MLR}) > 0$, per cui P contiene sicuramente membri MLR. Si può dire molto di più; se P è Π^0_1 contiene, infatti,rappresentanti per tutti i gradi MLR.

A tal proposito, ricordiamo alcune nozioni sui gradi di Turing. Se esiste una macchina di Turing capace di calcolare la funzione caratteristica dell'insieme Y usando come oracolo X, si scrive $Y \leq_T X$ e si dice che Y è *Turing-riducibile* a X. Se $Y \leq_T X$ e $X \leq_T Y$ si scrive $X \equiv_T Y$. La relazione $\equiv_T$ è di equivalenza; la classe di equivalenza di un certo X si chiama *grado di Turing* di X. Se A, B sono insiemi, identificati con le successioni infinite di bit che li rappresentano si definisce $A \oplus B$ come la successione (identificata a un insieme) che ha $A(n)$ nel posto $2n$ e $B(n)$ nel posto $2n+1$. Si vede subito che $A \oplus B$ è l'estremo superiore dei gradi di A e B rispetto all'ordine $\leq_T$ e che l'insieme dei gradi di Turing forma dunque un semireticolo.

In seguito, sarà usata anche una diversa nozione di riducibilità, quella di *riducibilità wtt.* Supponiamo che la computazione $\Phi^X_e(x)$ converga. Se n è il più grande numero per cui, durante la computazione, viene posta all'oracolo la domanda "$n \in X$?", il numero $n+1$ si dice *uso* di $\Phi^X_e(x)$ e si indica con $\mathrm{use}^X_e(x)$. Se Φ^X_e calcola la funzione caratteristica di Y e, inoltre, $\mathrm{use}^X_e(n) \leq f(n)$ per una qualche funzione computabile totale f, Y si dice $wtt{-}riducibile$ a X e si scrive $Y \leq_{wtt} X$. L'abbreviazione *wtt* sta per "weak table reducible".

Definizione 2.2.1. Dato un insieme di stringhe B, definiamo induttivamente gli insiemi B^n:

$$B^0 = B$$

$$B^{n+1} = \{\hat{\sigma\tau} \ : \ \sigma \in B^n\}$$

Osservazione 2.2.2. Chiaramente, $\lambda(B^n) = \lambda(B)^{n+1}$.

Ricordiamo che, date due successioni $Y, Z \in 2^{\mathbb{N}}$, si dice che Y è una *coda* di Z se esiste una stringa σ tale che $Z = \sigma Y$.

Proposizione 2.2.3. *Sia P una classe Π^0_1 di misura positiva. Allora, per ogni $Z \in MLR$, P contiene una coda di Z.*

Dimostrazione. Poniamo $A = 2^{\mathbb{N}}\ P$. Allora A è una classe Σ^0_1 con $\lambda(A) < 1$. Poniamo $A = [\![R]\!]$, con R insieme c.e. Definiamo, inoltre, $A^n = [\![R^n]\!]$. Per l'osservazione precedente e per la proposizione 1.1.14,

$$d(\sigma) = \sum_n d_{A^n}(\sigma)$$

è una martingala c.e. Poiché $Z \notin S[d]$, ragionando come nella seconda parte della dimostrazione del teorema 2.1.17, si conclude che esiste un k tale che $Z \notin A^k$. Ci sono due casi: o il minimo k per cui $Z \notin A^k$ è 0, o è maggiore di 0. Nel primo caso, $Z \notin A^1 = A$, per cui $Z \in P$. Nel secondo caso, $Z \in A^{k-1}$ ma $Z \notin A^k$. Consideriamo la più corta stringa $\sigma \prec Z$ tale che $\sigma \in R^{k-1}$. Si ha che, scrivendo $Z = \sigma Y$, nessun prefisso di Y sta in R, per cui $Y \notin A$, cioè $Y \in P$. $\square$

Poiché ogni stringa ha lo stesso grado di Turing delle sue code,si ottiene il risultato seguente.

Corollario 2.2.4. *Sia P una classe Π^0_1 di misura positiva. Allora per ogni $Z \in MLR$ esiste un $Y \in P$ che è Turing-equivalente a Z.*

2.3 Il teorema di Kučera

Il teorema di Kučera afferma che, usando come oracolo insiemi MLR, è possibile computare ogni altro insieme; si tratta di un risultato controintuitivo, perché dimostra che un sottoinsieme computabile di MLR codifica sempre una quantità arbitaria d'informazione. Come si vedrà dalla dimostrazione, ciò è legato strettamente al fatto che esistono classi Π^0_1 e di misura positiva formate da insiemi MLR (tale è, ad esempio, la classe $P = 2^{\mathbb{N}} - U_1$, se $\{U_n\}$ è un test ML universale).

Lemma 2.3.1. *Sia $S \subseteq 2^{\mathbb{N}}$ una classe misurabile e x una stringa. Supponiamo che $\lambda(S|x) \geq 2^{-(r+1)}$. Esistono, allora, stringhe distinte $y_0, y_1 \succeq x$ di lunghezza $|x| + r + 2$ tali che $\lambda(S|y_i) > 2^{-(r+2)}$, con $i = 0, 1$*

Dimostrazione. Osserviamo che la corrispondenza $Y \longrightarrow xY$ induce un isomorfismo fra lo spazio $2^{\mathbb{N}}$ dotato della misura di Lebesgue e lo spazio $[\![x]\!]$ dotato della misura di probabilità condizionata $\lambda(\bullet|x)$ Ricordiamo, a tal proposito, che due spazi di misura $(E, \mathcal{E}, \mu)$ e $(F, \mathcal{F}, \nu)$ si dicono isomorfi se esiste una funzione biiettiva $f : E \longrightarrow F$ misurabile e con inversa misurabile, tale che per ogni $A \in \mathcal{E}$, $\mu(A) = \nu(f(A))$. In virtù di questo isomorfismo, è sufficiente dimostrare il lemma nel caso che x sia la stringa vuota; per passare al caso in cui x non sia vuota, basterà applicare il lemma come se x fosse vuota e poi anteporre x alle stringhe y_0, y_1 così ottenute. Sia

$$y_0 = \arg\max_{y \in 2^{r+2}} \lambda(S|y)$$

Supponiamo che per ogni $y \neq y_0$ di lunghezza $r + 2$ si verifichi $\lambda(S|y) \leq 2^{-(r+2)}$. Allora avremmo:

$$\begin{aligned}
\lambda(S) &= \sum_{y \in 2^{r+2}} \lambda(S \cap [\![y]\!]) = \sum_{y \in 2^{r+2}} \lambda(S|y)\lambda([\![y]\!]) = \lambda(S|y_0)\lambda([\![y]\!]) + \\
&+ \sum_{y \in 2^{r+2} \wedge y \neq y_0} \lambda(S|y)\lambda([\![y]\!]) \leq 2^{-(r+2)} + (2^{(r+2)} - 1)2^{-(r+2)}2^{-(r+2)} \\
&< 2^{-(r+1)}.
\end{aligned}$$

Questo è assurdo perché abbiamo supposto $\lambda(S|x) = \lambda(S) \geq 2^{-(r+1)}$, per cui deve esistere $y_1 \in 2^{r+2}$ tale che $\lambda(S|y_1) > 2^{-(r+2)}$. Per come è definito y_0, $\lambda(S|y_0) \geq \lambda(S|y_1) > 2^{-(r+2)}$ □

Teorema 2.3.2 (Kučera)**.** *Sia Q una classe Π^0_1 non vuota formata da insiemi MLR. Per ogni $X \in 2^{\mathbb{N}}$ esiste uno $Z \in Q$ tale che*

$$X \leq_{wtt} Z \leq_{wtt} X \oplus \emptyset'$$

Dimostrazione. Diamo in primo luogo un'idea intuitiva della dimostrazione. Definiremo un albero T, il cui corpo $[T]$ sarà usato per codificare la stringa X. A ogni bit di X corrisponderà, infatti, la scelta di una certa diramazione di T e l'insieme di tutte queste scelte definirà un elemento nel corpo di T. Per l'esempio 2.1.3, $\lambda(Q) > 0$. Pertanto, esiste un $k \geq 0$ tale che $\lambda(Q) > 2^{-k-1}$. Sia f una funzione definita ricorsivamente come segue:

$$f(0) = k\ ,\ f(r + 1) = f(r) + r + 2$$

Definiamo ora un albero T:

$$T = \{\sigma \ : \ \forall r[f(r) \leq |\sigma| \ \longrightarrow \lambda(Q|(\sigma|_{f(r)})) \geq 2^{-(r+k+1)}]\}$$

Osserviamo dapprima che, siccome Q è una classe Π^0_1, ne esiste un'approssimazione effettiva Q_s con $Q_s \supseteq Q_{s+1}$ e $Q = \bigcap Q_s$, per cui $\lambda(Q|(\sigma|_{f(r)}))$ può essere approssimata dall'alto con $\lambda(Q_s|(\sigma|_{f(r)}))$ e la condizione che definisce T può essere scritta come:

$$\forall r \forall s[f(r) \leq |\sigma| \ \longrightarrow \lambda(Q_s|(\sigma|_{f(r)})) \geq 2^{-(r+k+1)}]$$

dal che si vede che $T \subseteq 2^{<\omega}$ è un albero Π^0_1. Sia ora $[T]$ la classe dei cammini attraverso T. Si vede per induzione, usando il lemma precedente, che tale classe è non vuota. Un risultato basilare assicura che $[T] \subseteq 2^{\mathbb{N}}$ è una classe Π^0_1 (si veda [5], pag. 53).
Diciamo che se U è una classe nello spazio di Cantor, si dice che una stringa *sta su* U se è prefisso di un elemento di U. Data una stringa τ, definiremo ora una stringa x_τ che sta su $[T]$ e codifica τ. Usiamo il seguente algoritmo:

- $x_\oslash = \oslash$
- supponiamo di aver definito x_τ
- sia $x_{\tau 0}$ quella stringa y che è minimale per l'ordinamento lessicografico fra quelle che stanno su $[T]$, hanno prefisso x_τ e lunghezza $f(|\tau| + 1)$.
- sia $x_{\tau 1}$ quella stringa y che è massimale per l'ordinamento lessicografico fra quelle che stanno su $[T]$, hanno prefisso x_τ e lunghezza $f(|\tau| + 1)$.

Osserviamo che esiste sempre una stringa y colla proprietà cercata in virtù del lemma precedente, il quale assicura anche che $x_{\tau 0}$ e $x_{\tau 1}$ sono distinte. Poniamo allora

$$Z = \sup_{\tau \prec X} x_\tau.$$

Osserviamo che, per poter trovare $Z(n)$, l'algoritmo ha bisogno di conoscere un segmento iniziale di X di lunghezza al più n (infatti, con tale segmento iniziale si può calcolare un segmento iniziale di Z di lunghezza $f(n+1) \geq n$) e, inoltre deve poter decidere se una certa stringa y stia su $[T]$; come verrà mostrato nel prossimo lemma, ciò si può fare tramite una riduzione *wtt* con oracolo $\emptyset'$. Da questo si vede che $Z \leq_{wtt} X \oplus \emptyset'$
Descriviamo ora un algoritmo che, funzionando con oracolo Z e uso al più $f(n + 1)$, calcola $X(n)$. Poniamo $x = Z|_{f(n)}$ e $y = Z|_{f(n+1)}$. Cerchiamo un'approssimazione $[T]_s$ (la quale esiste sempre per costruzione di Z) tale che:

- $[T]_s$ non contiene stringhe che estendano x, abbiano lunghezza pari a $|y|$ e precedano y nell'ordine lessicografico, oppure

- $\hat{[T]}_s$ non contiene stringhe che estendano x, abbiano lunghezza pari a $|y|$ e seguano y nell'ordine lessicografico

Nel primo caso, guardando la procedura di costruzione di Z, si vede che $X(n) = 0$, nel secondo $X(n) = 1$. □

Lemma 2.3.3. *Sia $\hat{Y}$ una classe Π^0_1. Sia Y l'insieme delle stringhe che stanno su $\hat{Y}$. Allora $Y \leq_{wtt} \emptyset'$ e la riduzione non dipende da $\hat{Y}$.*

Dimostrazione. Supponiamo $2^{\mathbb{N}} \setminus \hat{Y} = [\![W_e]\!]$. Da e si può ottenere effettivamente una funzione computabile g tale che:

$$\sigma \in W_e \Leftrightarrow g(\sigma) \in \emptyset'$$

(se ne veda la dimostrazione in [5], pag.9). Descriviamo allora una procedura per decidere se σ sta su $\hat{Y}$.

- Per ogni $\tau \preceq \sigma$, calcoliamo $g(\tau)$.

- Se, per ogni $\tau \preceq \sigma$, $g(\tau) \notin \emptyset'$, $[\![\sigma]\!]$ non è contenuto in $[\![W_e]\!]$, per cui σ sta su $\hat{Y}$

- Altrimenti, σ non sta su $\hat{Y}$.

Poniamo ora:

$$f(\sigma) = \max_{\tau \preceq \sigma} g(\sigma)$$

Si ha che $\mathrm{use}^{\emptyset'}(\sigma) \leq f(\sigma)$. □

2.4 Il concetto di stocasticità secondo von Mises-Church

Un test statistico intuitivo per stabilire se una sequenza di bit sia casuale o meno (rispetto alla probabilità uniforme) consiste nel verificare se la sequenza rispetti la legge dei grandi numeri, cioè se la frquenza di 0 e di 1 all'interno dei suoi segmenti iniziali tenda a $\frac{1}{2}$ quando la lunghezza di questi segmenti tende all'infinito; d'altra parte, è chiaro che molte sequenze che soddisfano questa proprietà non sono in alcun modo casuali, come, ad esempio la sequenza che ha 1 nei posti pari e 0 negli altri. Se però la legge dei grandi numeri vale per ogni sottosequenza che si può estrarre dalla sequenza data, si ottiene una nozione più plausibile di casualità. A questo è naturale aggiungere la richiesta che la regola di selezione della sottosequenza sia computabile. Si arriva così alla nozione di stocasticità secondo von Mises-Church.

Definizione 2.4.1. Una regola di selezione f è una funzione parzialmente computabile $f\ 2^{<\omega} \mapsto \{0,1\}$

Definizione 2.4.2. Dato un insieme Z e una regola di selezione f, definiamo $s_f(Z,n)$ come l'$n-$esimo numero k (se esiste) per cui $f(Z|_k) = 1$.

La regola di selezione permette dunque di estrarre dalla successione Z la sottosuccessione $Z_f = Z(s_f(Z,1))Z(s_f(Z,2))Z(s_f(Z,3))\ldots$.

Definizione 2.4.3. Se $s_f(Z,n)$ è definito, poniamo:

$$S_f(Z,n) = \sum_{k\leq n} Z(s_f(Z,k))$$

$S_f(Z,n)$ è, dunque il numero di 1 presenti nei primi n bit della sottosequenza Z_f.

Definizione 2.4.4. Una sequenza Z è stocastica [secondo von Mises-Church] se per per ogni regola di selezione f tale che Z_f ha lunghezza infinita si ha:

$$\lim_n \frac{(S_f(Z,n))}{n} = \frac{1}{2}$$

Teorema 2.4.5. *Se Z è MLR, è stocastica nel senso di von Mises-Church. Il viceversa non è vero.*

Dimostrazione. Consideriamo una regola di selezione f. A partire da essa costruiremo una martingala inferiormente semicomputabile che ha successo su Z se Z_f non rispetta la legge dei grandi numeri, dal che si dedurrà che se Z è MLR, ogni sottosuccessione Z_f soddisfa la legge dei grandi numeri. Procediamo per gradi; in primo luogo, per ogni naturale k definiamo due martingale computabili d_k^0 e $d_k^!$ per induzione sulla lunghezza della stringa σ.

1. $d_k^0(\oslash) = d_k^1(\oslash) = 1$
2. Se $f(\sigma)$ non è definita su σ,

   $$d_k^0(\sigma j) = d_k^1(\sigma j) = 0$$

   per $j = 0,1$

3. Se $f(\sigma) = 0$,
$$d_k^0(\sigma j) = d_k^0(\sigma)$$
e
$$d_k^1(\sigma j) = d_k^1(\sigma)$$
per $j = 0, 1$

4. Se $f(\sigma) = 1$:
$$d_k^0(\sigma 0) = (1 + 2^{-k})d_k^0(\sigma)$$
e
$$d_k^0(\sigma 1) = (1 - 2^{-k})d_k^0(\sigma)$$
mentre:
$$d_k^1(\sigma 0) = (1 - 2^{-k})d_k^1(\sigma)$$
e
$$d_k^1(\sigma 1) = (1 + 2^{-k})d_k^1(\sigma)$$

E' molto facile capire quale sia la strategia rappresentata da d_k^0: aspettiamo che f scelga un bit di Z e puntiamo una quota pari a $1/2^k$ del nostro capitale sul fatto che tale bit sia 0. Verifichiamo che d_k^0 sia davvero una martingala (il fatto che sia computabile è ovvio e deriva interamente dal carattere computabile della f). Se $f(\sigma)$ non è definita,
$$d_k^0(\sigma 0) + d_k^0(\sigma 1) = 2d_k^0(\sigma).$$
Se $f(\sigma) = 1$,
$$d_k^0(\sigma 0) + d_k^0(\sigma 1) = (1 + 2^{-k})d_k^0(\sigma) + (1 - 2^{-k})d_k^0(\sigma) = 2d_k^0(\sigma)$$
Considerazioni del tutto analoghe valgono per d_k^1 e per la strategia che quest'ultima rappresenta. Ora definiamo:
$$d(\sigma) = \sum_k 2^{-k}(d_k^0(\sigma) + d_k^1(\sigma)).$$
Tale d è chiaramente una martingala inferiormente semicomputabile per 2.1.14. Mostriamo che se la successione Z_f è infinita e non rispetta la legge dei grandi numeri, $Z \in S[d]$, il che implica subito la tesi. Fissiamo alcune notazioni: poniamo $n_1(n) = S_f(n, Z)$ e $n_0(n) = n - S_f(n, Z)$, cioé $n_i(n)$ è uguale al numero di bit uguali a i in $Z_f|_n$. Ora supponiamo che Z_f non rispetti la legge dei grandi numeri; dato $\epsilon > 0$, non è restrittivo supporre che esistano infiniti n per cui $n_0(n) > (1/2 + \epsilon)n$ (se così non fosse, esisteranno infiniti n per cui $n_1(n) > (1/2 + \epsilon)n$ e per simmetria si procede in modo

simile).
Poniamo $m(n) = s_f(n, Z)$, cioè m è il posto immediatamente successivo all' $n-$esimo bit scelto da f. Allora per ogni k e per ogni n tale che $n_0(n) > (1/2 + \epsilon)n$:

$$d_k^0(Z|m(n)) = (1 + 2^k)^{n_0(n)}(1 - 2^k)^{n_1(n)} \geq (1 + 2^k)^{\frac{1}{2}n+\epsilon}(1 - 2^k)^{\frac{1}{2}n-\epsilon}$$

Passando ai logaritmi:

$$\log d_k^0(Z|m(n)) \geq n((\frac{1}{2} + \epsilon)\log(1 + 2^k) + (\frac{1}{2} - \epsilon)\log(1 - 2^k))$$

Consideriamo ora la funzione $h(x) = (\frac{1}{2} + \epsilon)\log(1 + x) + (\frac{1}{2} - \epsilon)\log(1 - x)$. Essa è differenziabile intorno allo 0 e in questo punto vale 0, mentre la sua derivata in 0 vale 2ϵ, per cui c'è un $a > 0$ tale che $h(x) > 0$ se $x \in]0, a[$. Dunque, a patto di scegliere k in modo che $2^{-k} < a$, si ha che :

$$\log d_k^0(\sigma) \geq \delta n$$

per $\delta = h(2^{-k})$ che non dipende da n, cioè:

$$d_k^0(\sigma) \geq (2^\delta)^n.$$

Quando n tende a $+\infty$, si vede che $Z \in S[d_k^0] \subseteq S[d]$.
Il significato intutivo della dimostrazione è elementare: se la frequenza di 0 rimane significativamente superiore a quella degli 1, è possibile vincere somme sempre più grandi puntando sullo 0.
Che il viceversa non valga è il contenuto del prossimo teorema e dell'osservazione che lo segue. $\square$

Teorema 2.4.6 (Ville). *Sia E una famiglia di regole di selezione (che qui consideriamo essere funzioni parziali non necessariamente computabili). Allora esiste una sequenza X tale che:*

1. $$\lim_n \frac{S_f(n, X)}{n} = \frac{1}{2}$$
   per ogni f tale che X_f è infinta.

2. *Per ogni n*
   $$\frac{S_i(f, X)}{n} < \frac{1}{2}$$
   dove i indica la regola di selezione che vale costantemente 1 (cioè $X_f = X$).

Dimostrazione. Non riportiamo la dimostrazione perché basata su considerazioni combinatorie e in nulla legate alla teoria della computabilità. Si veda, ad esempio, [6], paragrafo 6.5. □

Osservazione 2.4.7. La successione X la cui esistenza è garantita dal teorema di Ville è stocastica nel senso di Von Mises-Church, ma non è MLR. Per vederlo, basta trovare una classe Π^0_1 di misura nulla che la contenga. Tale classe è la seguente:

$$\{X : \frac{S_i(n, X)}{n} \leq \frac{1}{2}\}.$$

Che essa contenga X è banale; inoltre è Π^0_1 perché il suo complementare si può scrivere come segue:

$$\{X : \exists n[\frac{S_i(n, X)}{n} > \frac{1}{2}]\}$$

e questa è chiaramente una classe Σ^0_1 per via della caratterizzazione ricordata all'inizio di questo capitolo.

Capitolo 3

Dimensione di Hausdorff effettiva

3.1 Casualità per la misura di Hausdorff

Ricordiamo che, se h è una funzione d'ordine, con H^h s'indica la misura di Hausdorff associata e con ρ_h la corrispondente premisura. Nel seguito, assumeremo sempre che h sia computabile.
Osserviamo che h assume valori reali, per cui è necessaria una diversa definizione di computabilità; nel seguito, useremo la seguente:

Definizione 3.1.1. $f : \mathbb{N} \mapsto \mathbb{R}$ si dice *computabile* se esiste una funzione computabile $h(n, k)$ tale che per ogni n e k:

$$|f(n) - h(n, k)| < \frac{1}{2^n}$$

Inoltre, se $s \in [0, 1]]$, H^s è la misura di Hausdorff associata alla funzione $x \mapsto sx$ e, come al solito, ρ_s la premisura corrispondente. Se proviamo a dare una versione effettiva del concetto di "classe di misura di Hausdorff nulla", risulta naturale pensare alla condizione data dal lemma 1.2.5 e dare, pertanto, la definizione seguente:

Definizione 3.1.2. Un h-test [semplice] di Martin-Löf (h-Martin-Löf test, test h-ML) è una successione $\{V_n\}$ di insiemi di stringhe uniformemente c.e. tali che:

$$\sum_{\sigma \in V_n} \rho_h(\sigma) \leq 2^{-n} \tag{3.1}$$

Se $s \in [0, 1]$, per s-test di Martin-Löf s'intende un test rispetto alla funzione d'ordine $h(x) = sx$

Definizione 3.1.3. Uno $Z \in 2^{\mathbb{N}}$ si dice h-Martin-Löf random (h-MLR) se per nessun $h-$test si ha:

$$Z \in \bigcap [\![V_n]\!]$$

E' allora naturale emulare la definzione di dimensione di Hausdorf:

Definizione 3.1.4.

$$\dim^1_H(Z) = \inf\{s \in [0,1] \cap \mathbb{Q} \ : \ Z \text{ non è } s\text{-Martin-Löf random}\}$$

Osservazione 3.1.5. Sia $s \in [0,1]$ $0 < r < s$. Allora:

$$\sum_{\sigma \in V_n} 2^{-s|\sigma|} = \sum_{\sigma \in V_n} 2^{-(s-r+r)|\sigma|} \leq \sum_{\sigma \in V_n} 2^{-(s-r)|\sigma|}$$

Dunque, se $\{V_n\}$ è un test t-ML, è anche un test s-ML per ogni $t < s < 1$. Pertanto, se Z non è t-ML random, non è s-ML random per alcun $t < s < 1$. Viceversa, se è t-ML random, è s-ML random per ogni $0 < s < t$. Abbiamo dunque la situazione seguente:

$$\begin{cases} Z \text{ è } s\text{-ML random} & \text{se s} < \dim(Z) \\ Z \text{ non è } s\text{-ML random} & \text{se s} > \dim(Z) \\ \text{Non si può concludere alcunché} & \text{se s} = \dim(Z) \end{cases}$$

Ritroviamo una caratterizzazione della casualità in termini di complessità del segmento iniziale:

Definizione 3.1.6. Z si dice h-complesso se esiste una costante $c \geq 0$ tale che

$$K(Z|_n) \geq h(n) - c$$

Proposizione 3.1.7. *Z è h-ML random se e solo se è $h-$complesso*

Dimostrazione. E' molto simile a quella di 2.1.6. Se Z non è h-complesso, gli insiemi

$$R_c = \{x \in 2^{<\omega} \ : \ K(x) \leq h(x) - c\}$$

formano, al variare di c, un test h-ML. Viceversa, se $\{V_n\}$ è un test h-ML tale che $Z \in \cap [\![V_n]\!]$, le condizioni:

$$\{\langle h(\sigma) - k, \sigma \rangle \quad \sigma \in U_{2k}\}$$

dove U_n è un sottoinsieme senza prefissi di V_n che genera la stessa foresta, formano un'insieme accettabile di richieste, per cui $\forall k \exists n_k \ : \ K(Z|_{n_k}) \leq h(Z|_{n_k}) - k$. $\square$

Vorremmo ora trovare una caratterizzazione tramite supermartingale; diamo, in primo luogo, una definizione:

Definizione 3.1.8. Sia $d : 2^{\mathbb{N}} \mapsto \mathbb{R}^{\geq 0}$ una supermartingala e o una funzione. L'insieme degli o-successi di d si definisce come segue:

$$S_o[d] = \{Z \ : \ \limsup_{n\to\infty} \frac{d(Z|_n)}{o(n)} = +\infty\} \tag{3.2}$$

Sembra plausibile che, emulando la dimostrazione fatta per la misura di Lebesgue, si riesca a mostrare che Z è h-ML random se e solo per ogni supermartingala d inferiormente semicomputabile $Z \notin S_{2^{n-h(n)}}[d]$; risultato che coincide, per $h(n) = 1n$, con quello che vale per la casualità ML usuale. Purtroppo, un'analisi degli argomenti usati in quest'ultimo caso evdenzia un ostacolo insormontabile. Data una supermartingala d, avevamo definito nella prova di 2.1.17 gli insiemi $\{\sigma \ : \ d(\sigma) \geq 2^n\}$, allo scopo di mostrare che generano un test ML. Ora potremmo, similmente, definire insiemi $V_k = \{\sigma \ : d(\sigma) \geq 2^{|\sigma|-h(|\sigma|)}2^n\}$. A questo punto, però, dovremmo provare che:

$$\sum_{\sigma\in V_n} 2^{-h(|\sigma|)} \leq 2^{-n}$$

In generale, l'insieme V_n non è un anticatena, per cui è inverosimile che la disuguaglianza possa valere. A suo tempo, avevamo usato la disugualglianza di Kolmogorov 2.1.11,che può essere enunciata come segue:

Proposizione 3.1.9. *(I) Sia data un'anticatena C di estensioni di una certa stringa σ; vale allora:*

$$\sum_{\tau\in C} 2^{-|\tau|}d(\tau) \leq 2^{-|\sigma|}d(\sigma)$$

(II) Sia $R_k \subseteq \{\sigma \ : \ d(\sigma) \geq k\}$ un'anticatena. Allora $\sum_{\sigma\in R_k} 2^{-|\sigma|} \leq \frac{d(\emptyset)}{k}$

D'altra parte, se pensiamo di ricavare in modo effettivo da V_n un'anticatena C_n che generi lo stesso test, risulta ugualmente difficile garantire che la disuguaglianza valga per quest'ultima. Infatti, supponiamo che, nell'enumerazione di V_n s'incontrino, in quest'ordine, le stringhe $110, 0111, 11$. Dapprima enumeriamo 110 in C_n, senza poter sapere che a un passo successivo troveremo 11. Quando ciò capita, siamo costretti a mettere in C_n la stringa 111, ma abbiamo che $2^{-s|110|} + 2^{-s|111|} = 2^{-3s+1} > 2^{-2s} = 2^{s|11|}$. Da questo se vede che in generale la sommatoria $\sum_{\sigma\in V_n} 2^{-h(|\sigma|)}$ potrebbe crescere indefinitamente. Appare, a questo punto, opportuno dare un diverso concetto di casualità, che sia più facile da caratterizzare tramite supermartingale. Il seguente approccio è presente in [3].

Definizione 3.1.10. Un h-test forte (strong h-Martin-L"of test, test h-SML) è una successione V_n di insiemi di stringhe uniformemente c.e tali che, per ogni anticatena $\hat{V}_n \subseteq V_n$:

$$\sum_{\sigma \in \hat{V}_n} \rho_h(\sigma) \leq 2^{-n} \tag{3.3}$$

Se $s \in [0,1] \cap \mathbb{Q}$, per s-test forte di Martin-Löf s'intende un test rispetto alla funzione d'ordine $h(x) = sx$

Definizione 3.1.11. Uno $Z \in 2^{\mathbb{N}}$ si dice fortemente h-Martin-Löf random (h-SMLR) se per nessun $h-$test si ha:

$$Z \in \bigcap [\![V_n]\!]$$

Teorema 3.1.12. *Sia s razionale. Allora le condizioni seguenti si equivalgono:*
(I)Z è fortemente s-Martin-Löf random
(II)per ogni supermartingala inferiormente c.e. d,

$$Z \notin S_{2^{n-sn}}[d]$$

Dimostrazione. Nel seguito della dimostrazione, $h(n) := sn$. Supponiamo, dapprima, che valga (I). Sia d una supermartingala inferiormente c.e. Non è restrittivo supporre che $d(\oslash) \leq 1$ (si veda 2.1.17). Definamo:

$$V_n = \{x \ : \ 2^{-|x|}d(x) > 2^{-h(|x|)}2^n\}$$

Se $X \in S_{2^{n-h(n)}}[d]$,

$$\limsup_{n\to\infty} \frac{d(X|_n)}{2^n 2^{-h(n)}} = \infty$$

per cui per ogni n esiste una M tale che $X|_M \in V_n$, sicché $X \in \cap_n [\![V_n]\!]$. Se riusciamo a provare che i V_n formano un h- test forte, siamo arrivati a una contraddizione e abbiamo finito. Ma se $\hat{V}_n \subseteq V_n$ è un'anticatena, abbiamo:

$$\sum_{\sigma\in\hat{V}_n} \rho_h(\sigma) = \sum_{\sigma\in\hat{V}_n} 2^{-h(|\sigma|)} \leq \frac{1}{2^n} \sum_{\sigma\in\hat{V}_n} 2^{-|\sigma|}d(\sigma) \leq \frac{1}{2^n}$$

dove l'ultimo passaggio deriva dalla disugluaglianza 1 in 3.1.9. Rimane daprovare che i V_n sono uniformemente c.e, ma questo si fa esattamente come in 2.1.17.
Supponiamo ora che valga (II). Sia V_n un h-test forte. Per ogni n, definiamo dapprima una famiglia monotona $d_{n,s}$ di supermartingale computabili.

$$d_{n,s}(\sigma) = n \max\{2^{|\sigma|} \sum_{x\in C} 2^{-h(|x|)} \ : \ C \subseteq V_{n,s} \cap \sigma 2^{<\omega} \text{ anticatena}\}$$

Allora :

$$\begin{aligned}
&d_{n,s}(\sigma 0) + d_{n,s}(\sigma 1) = \\
n \max\{2^{|\sigma|+1} &\sum_{x \in C} 2^{-h(|x|)} \ : \ C \subseteq V_{n,s} \cap (\sigma 0) 2^{<\omega} \text{ anticatena}\} \\
+n \max\{2^{|\sigma|+1} &\sum_{x \in C} 2^{-h(|x|)} \ : \ C \subseteq V_{n,s} \cap (\sigma 1) 2^{<\omega} \text{ anticatena}\} \\
\leq n \max\{2^{|\sigma|+1} &\sum_{x \in C} 2^{-h(|x|)} \ : \ C \subseteq V_{n,s} \cap \sigma 2^{<\omega} \text{ anticatena}\} \\
&= 2d(\sigma)
\end{aligned}$$

dove l'ultima disuguaglianza dipende da fatto che dati $C_0 \subseteq V_{n,s} \cap (\sigma 0) 2^{<\omega}$ anticatena e $C_1 \subseteq V_{n,s} \cap (\sigma 1) 2^{<\omega}$ anticatena, C_0 e C_1 sono disgiunti e la loro unione è un'anticatena contenuta in $V_{n,s} \cap \sigma 2^{<\omega}$. Pertanto, $d_{n,s}$ è chiaramente una supermartingala. Siccome $V_{n,s} \subset V_{n,s+1}$, la successione $d_{n,s}$ è monotona non decrescente in s. Poniamo allora:

$$d_n(\sigma) = \sup_s d_{n,s}(\sigma)$$

La funzione d_n è chiaramente una supermartingala c.e, perché le disuguaglianze

$$d_{n,s}(\sigma 0) + d_{n,s}(\sigma 1) \leq 2 d_{n,s}(\sigma)$$

passano al limite ed è c.e perché estremo superiore di funzioni computabili. Se $Z|_{k_n} \in V_n$,esiste un s tale che $Z|_{k_n} \in V_{n,s}$, per cui $d_s(Z|_{k_n}) \geq n 2^{k_n} 2^{-h(k_n)}$. Per vederlo, si ponga $C = \{Z|_{k_n}\} \subseteq V_{n,s} \cap (Z|_{k_n}) 2^{<\omega}$. Ponendo $d = \sum_n d_n$, se vede subito che $Z \in S_{2^{n-h(n)}}[d]$. □

Osservazione 3.1.13. La dimostrazione precedente può essere estesa immediatamente a tutte le funzioni tali che d'ordine h tali che $n > h(n)$ da un certo n in poi.

Ricordiamo che KA indica la complessità a priori.

Corollario 3.1.14. *Z è fortemente s-Martin-Löf random se e solo se esiste una costante $c \geq 0$ tale che:*

$$KA(Z|_n) \geq sn - c \qquad \forall n \tag{3.4}$$

Dimostrazione. Basta ricordare che $KA(\sigma) = |\sigma| - \log_2 d(\sigma)$ per una semimartingala ottimale d. Poiché Z è fortemente s-MLR, $Z \notin S_{2^{n-sn}}[d]$, per cui esiste una costante c tale che $d(X|n) \leq c 2^{-n} 2^{sn}$. Passando ai logaritmi si ha:

$$\log d(X|n) \leq c \log(2^{-n} 2^{sn})$$

da cui la tesi. □

Si pone ora in modo naturale la questione di indagare il legame fra h-test ML e h-test forti. Evedentemente, ogni h-test debole è anche un h-test forte, per cui se un insieme Z è fortemente h-ML, è anche h-ML. Il teorema 13.5.12 in [6] afferma che, d'altra parte, quando $h(n) = sn$, $s \in [0, 1[\cap\mathbb{Q}$, esistono insiemi che sono s-ML random, ma non fortemente s-ML random. La nozione di randomness forte è, pertanto, realmente più forte di quella di randomness semplice. Per $s = 1$, d'altra parte, le due nozioni si equivalgono e coincidono colla nozione di casualità ML introdotta nel capitolo 2. Quest'ultima risulta, infatti, essere nient'altro che la randomness forte nel caso della premisura $\rho_1(\sigma) = 2^{-|\sigma|}$. Vediamo, ora, una procedura effettiva che permette di trasformare un 1-test forte in un 1-test semplice. Siano dati gli insiemi uniformemente c.e. V_n colla proprietà che per ogni anticatena $\hat{V}_n \subseteq V$:

$$\sum_{\sigma\in\hat{V}_n} \rho_h(\sigma) \leq 2^{-n}$$

Enumeriamo ora gli elementi di V_n in una nuova classe U_n nel modo che segue:

1. $U_0 = \emptyset$
2. Se $y \in V_{n,s}$ è inconfrontabile cogli elementi di $U_{n,s-1}$, lo mettiamo in $U_{n,s}$
3. Se $y \in V_{n,s}$ è ha per prefisso un elemento di $U_{n,s-1}$, non lo mettiamo $U_{n,s}$
4. Se $y \in V_{n,s}$ è prefisso di un elemento $y\sigma$ di $U_{n,s-1}$, mettiamo in $U_{n,s}$ tutti gli elementi del tipo $y\tau$, con $|\tau| = |\sigma|$

Tale costruzione assicura che U_n è un'anticatena e che $[\![U_n]\!] = [\![V_n]\!]$; pertanto,

$$\sum_{\sigma\in U_n} 2^{-|\sigma|} = \lambda([\![U_n]\!]) = [\![(V_n]\!]) = \sum_{\sigma\in\hat{V}_n} \rho_h(\sigma)$$

per una certa anticatena $\hat{V}_n \subseteq V_n$ Inoltre, gli U_n, essendo ottenuti con una procedura effettiva da un'enumerazione dei V_n, sono uniformemente c.e.

Proposizione 3.1.15. *Supponiamo che Z sia s-ML random per un certo $s \in [0, 1] \cap \mathbb{Q}$. Allora, Z è fortemente t-ML random per ogni $t \in [0, s[\cap\mathbb{Q}$.*

Dimostrazione. Supponiamo Z non sia fortemente t-ML random per un certo $t < s$ razionale; sia V_n un s-test forte tale che $Z \in \cap_n[\![V_n]\!]$. Indichiamo con

2^k l'insieme delle stringhe di lunghezza k, che è chiaramente un'anticatena. Allora:

$$\sum_{\sigma \in V_n} 2^{-s|\sigma|} = \sum_k 2^{(t-s)k} \sum_{\sigma \in V_n \cap 2^k} 2^{-tk} \leq 2^{-n} \sum_k 2^{(t-s)k}$$

Ma $2^{(t-s)} < 1$ per cui la serie $\sum_k 2^{(t-s)k}$ converge a un reale a. Sia $2^m geqa$; allora ,ponendo $U_n = V_{n+m}$ si ottiene:

$$\sum_{\sigma \in U_n} 2^{-s|\sigma|} =\leq 2^{-n-m}a \leq 2^{-n-m}2^m = 2^n$$

da cui deriva che U_n è un $s-$test ML semplice, per cui Z non sarebbe $s-$ML random, ma questo è assurdo. □

3.2 Dimensione di Hausdorff e complessità

Daremo ora una caratterizzazione della dimensione di Hausdorff in termini di complessità del segmento iniziale. Seguiremo [4] e [6].

Proposizione 3.2.1.

$$\dim_H^1(A) = \liminf \frac{K(A|_n)}{n}$$

Lemma 3.2.2.

$$\dim_H^1(A) \geq \liminf \frac{K(A|_n)}{n}$$

Dimostrazione. Sia $s > \dim_H^1(A)$ razionale. Allora Z è s'-ML random per $s < s' < \dim_H^1(A)$, per cui è fortemente s-ML random e, per 3.1.14, $KA(Z|_n) \geq sn + O(1)$. Ma per [1], teorema 79:

$$K(\sigma) \leq KA(\sigma) + 2\log|\sigma| + O(1)$$

per cui:

$$\limsup \frac{sn - K(Z|_n)}{n} \geq \limsup \frac{sn - KA(Z|_n) - 2\log n}{n} \geq 0$$

per cui:

$$\limsup \frac{K(Z|_n)}{n} \leq s \qquad \forall s > dim_H^1(Z)$$

□

Lemma 3.2.3.

$$\dim_H^1(A) \leq \liminf \frac{K(A|_n)}{n}$$

Dimostrazione. Siano $r > s > \limsup \frac{K(Z|_n)}{n}$, r, s razionali. Definiamo:

$$B = \{\sigma \,:\, K(\sigma) \leq s|\sigma|\}$$

Definamo:

$$d(\sigma) = 2^{(1-s)|\sigma|}(\sum_{\sigma\tau \in B} 2^{-s|\tau|} + \sum_{\nu \in B \wedge \nu \prec \sigma} 2^{(s-1)(|\sigma|-|\nu|)})$$

Si verifica con un calcolo diretto che d è ben definita; inoltre è una martingala. Dividiamo i casi $\sigma \in B$ e $\sigma \notin B$. Se $\sigma \notin B$,

$$2^{(1-s)|\sigma|} \sum_{\sigma\tau \in B} 2^{-s|\tau|} = 2^{(1-s)|\sigma|} \sum_{\sigma 0\tau \in B} 2^{-s(|\tau|+1)} + 2^{(1-s)|\sigma|} \sum_{\sigma 1\tau \in B} 2^{-s(|\tau|+1)} =$$
$$2^{(1-s)|\sigma|} 2^{-s} \sum_{\sigma 0\tau \in B} 2^{-s|\tau|} + 2^{(1-s)|\sigma|} 2^{-s} \sum_{\sigma 1\tau \in B} 2^{-s|\tau|} =$$
$$\frac{1}{2}(2^{(1-s)|\sigma 0|} 2^{-s} \sum_{\sigma 0\tau \in B} 2^{-s|\tau|} + 2^{(1-s)|\sigma 1|} 2^{-s} \sum_{\sigma 1\tau \in B} 2^{-s|\tau|})$$

e lo stesso procedimento funziona per il secondo addendo. Se $\sigma \in B$:

$$2^{(1-s)|\sigma|} \sum_{\sigma\tau \in B} 2^{-s|\tau|} =$$
$$2^{(1-s)|\sigma|} \sum_{\sigma 0\tau \in B} 2^{-s(|\tau|+1)} + 2^{(1-s)|\sigma|} \sum_{\sigma 1\tau \in B} 2^{-s(|\tau|+1)} + 2^{(1-s)|\sigma|} =$$
$$\frac{1}{2}(2^{(1-s)|\sigma 0|} 2^{-s} \sum_{\sigma 0\tau \in B} 2^{-s|\tau|} + 2^{(1-s)|\sigma 1|} 2^{-s} \sum_{\sigma 1\tau \in B} 2^{-s|\tau|} + 2^{(1-s)|\sigma|})$$

ma il termine $2^{(1-s)|\sigma|}$ si cancella col secondo addendo, per cui la proprietà di martingala è ancora soddisfatta. Abbiamo che, quando $\sigma \in B$, $d(\sigma) \geq 2^{(1-s)|\sigma|}$. Inoltre, è inferiormente semicomputabile perché B è un insieme c.e. (per elencarlo, basta prendere una macchina senza prefissi U, calcolare parallelamente i vari $U(\tau)$ e vedere se l'output ha lunghezza maggiore di $s|\tau|$). Poiché $s > \limsup \frac{K(Z|_n)}{n}$, l'insieme $B \cap \{Z|_n \,:\, n \in \mathbb{N}\}$ è infinito, ma ciò significa che:

$$\limsup \frac{d(Z|_n)}{2^{(1-s)n}} = +\infty$$

per cui Z è fortemente s-random e $\dim^1_H(Z) \leq s$ □

Il teorema precedente permette di calcolare la dimensione di Hausdorff di un insieme Z; lo applicheremo per esibire, per ogni s razionale, due esempi di insiemi di dimensione effettiva s.

Esempio 3.2.4. Sia U una macchina di Turing universale senza prefissi; generalizzando una costruzione dovuta a Chaitin, poniamo:

$$\Omega^s = \sum_{U(\sigma)\downarrow} 2^{-\frac{|\sigma|}{s}}$$

Identifichiamo tale reale colla sua espansione decimale. Mostriamo che $\dim_H^1(\Omega^s) = s$. In virtù del teorema precedente, basterà mostrare che $K(\Omega^s|_n) = sn \pm O(1)$. Definiamo l'insieme finito $X_n = \{\sigma : |\sigma| \leq sn \wedge U(\sigma)\downarrow\}$. Conoscendolo, si possono elencare tutti gli elementi di complessità $\leq sn$ e individuare dunque un y con $K(y) \geq sn$; detta c una descrizione di quest'insieme, si ha dunque $sn \leq K(y) \leq K(c)+O(1)$. Ma X_n può essere calcolato effettivamente conoscendo $\Omega^s|_n$; basta applicare il seguente algoritmo:

1. Sia $k = 1$ e $a = 0$
2. Calcoliamo $U_s(\sigma)$ per tutti i t e i σ tali che $t+|\sigma| = k$; la computazione termina in un tempo finito.
3. Per ogni σ per cui nel passo precedente abbiamo trovato $U_t(\sigma)\downarrow$ (poiché U è senza prefissi, ciò significa che $U(\sigma)\downarrow$), poniamo $a = a + 2^{-|\sigma|/s}$ e, se $|\sigma| \leq sn$, mettiamo σ in X_n. Se $\Omega^s|_n < a$, l'algoritmo termina
4. Poniamo $k = k + 1$ e torniamo a 2

Supponiamo, per assurdo, che l'algoritmo precedente termini senza aver enumerato tutto X_n; vi sarebbe allora un ulteriore p tale che $|p| \leq sn$ e $U(p)\downarrow$. Ma allora, $\Omega^s \geq a + 2^{-|p|/s} \geq a + 2^{-n} > (\Omega^s|_n) + 2^{-n}$, il che è assurdo. Pertanto, c si ottiene effettivamente da Ω^s, per cui:

$$sn \leq K(c) + O(1) \leq K(\Omega^s|_n) + O(1)$$

Viceversa, conoscendo le prime $\lceil sn \rceil$ cifre di Ω^1 si può determinare in modo effettivo $\Omega^s|_n$. Infatti, chiamiamo b il razionale avente per espansione binaria la stringa formata dalle prime $\lceil sn \rceil$ cifre di Ω^1. Conoscendo quest'ultima, possiamo enumerare stringhe p_i fino ad avere:

$$\sum_{i=1}^{k} 2^{-|p_i|} > b$$

cioé

$$\sum_{i>k} 2^{-|p_i|} = \Omega^1 - b < 2^{-\lceil sn \rceil} \leq 2^{-sn}$$

Abbiamo allora:

$$\Omega^s - \sum_{i=1}^{k} 2^{-\frac{|p_i|}{s}} = \sum_{i>k}^{k} \frac{1}{2}^{\frac{|p_i|}{s}} \leq (\sum_{i>k}^{k} \frac{1}{2}^{|p_i|})^{\frac{1}{s}} < 2^n$$

dove abbiamo usato la disuguaglianza $x^q + y^q \leq (x+y)^q$, valida per $q > 1$ e $x, y \geq 0$. Allora è evidente che $\sum_{i=1}^{k} 2^{-\frac{|p_i|}{s}}$ è sufficiente a determinare in modo effettivo $\Omega^s|_n$; pertanto, $K(\Omega^s|_n) \leq K(\Omega^1|_{\lceil sn \rceil}) \leq sn + 2\log sn + O(1)$ Questa generalizzazione del numero di Chaitin viene introdotta in [9], dove si considera il caso più generale di di s computabile.

Esempio 3.2.5. Sia Z un insieme MLR (ad esempio, Ω^1). Definiamo un insieme Z^s come segue:

$$Z^s(n) = \begin{cases} Z(m) & \text{se } n = \lfloor m/s \rfloor \\ 0 & \text{altrimenti} \end{cases}$$

Allora, $Z^s|_k$ può essere calcolato effettivamente da $Z|_{\lceil sk \rceil}$ e, d'altra parte $Z|_{\lfloor ks \rfloor}$ può essere calcolato da $Z^s|_k$. Poiché Z è MLR, $K(Z|_n) \geq n - c$, pertanto:

$$ks + O(1) \leq K(Z|_{\lfloor ks \rfloor}) + O(1) \leq K(Z^s|_k)$$

D'altra parte,

$$ks + 2\log ks + O(1) \geq K(Z|_{\lceil sk \rceil}) \geq K(Z^s|_k)$$

Da questo, in virtù del teorema precedente si deduce che:

$$\dim_H^1(Z^s) = s$$

Bibliografia

[1] N. Vereschchagin A. Shen V.A. Uspenski. *Kolmogorov Complexity and Algorithmic Randomness*. Vol. 220. Mathematical Surveys and Monographs. American Mathematical Society, 2017.

[2] C.A.Rogers. *Hausdorff measures*. Cambridge University Press, 1970.

[3] S.A.Terwijn C.S.Calude L.Staiger. «On partial randomness». In: *Annals of Pure and Applied Logic* 138 (2006), pp. 20–30.

[4] E. Mayordomo. «A Kolmogorov Complexity characterization of constructive Hausdorff dimension». In: *Information Processing Letters* 84 (2002).

[5] A. Nies. *Computability and Randomness*. Oxford Science Publications, 2009.

[6] D.R. Hirschfeldt R.G. Downey. *Algorithmic Randomness and Complexity*. Springer-Verlag, 2010.

[7] R.L.Schilling. *Measures,Integrals and Martingales*. Cambridge University Press, 2005.

[8] J. Reimann. «Randomness-Beyond Lebesgue measure». In: *Proceedings of Logic Colloquium*. Cambridge University Press, 2006.

[9] K. Tadaki. «A generalization of Chaitin's halting probability Omega and halting self-similar sets». In: *Hokkaido Math. J.* 31.1 (2002).

[10] V.S.Borkar. *Probability Theory An Advanced Course*. Springer-Verlag, 1995.