Lecture notes on descriptional complexity and randomness

Peter Gács
Boston University
gacs@bu.edu

A didactical survey of the foundations of Algorithmic Information Theory. These notes are short on motivation, history and background but introduce some of the main techniques and concepts of the field.

The "manuscript" has been evolving over the years. Please, look at "Version history" below to see what has changed when.

Contents

Version history

June 2013: corrected a slight mistake in the section on the section on randomness via algorithmic probability.

February 2010: chapters introduced, and recast using the memoir class.

April 2009: besides various corrections, a section is added on infinite sequences. This is not new material, just places the most classical results on randomness of infinite sequences before the more abstract theory.

January 2008: major rewrite.

- Added formal definitions throughout.
- Made corrections in the part on uniform tests and generalized complexity, based on remarks of Hoyrup, Rojas and Shen.
- Rearranged material.
- Incorporated material on uniform tests from the work of Hoyrup-Rojas.
- Added material on class tests.

1 Complexity

Je n'ai fait celle-ci plus longue que parce que je n'ai pas eu le loisir de la faire plus courte.

Pascal

Ainsi, au jeu de *croix ou pile*, l'arrivée de croix cent fois de suite, nous paraît extraordinaire; parce que le nombre presque infini des combinaisons qui peuvent arriver en cent coups étant partagé en séries regulières, ou dans lesquelles nous voyons régner un ordre facile à saisir, et en séries irregulières; celles-ci sont incomparablement plus nombreuses.

Laplace

1.1 Introduction

The present section can be read as an independent survey on the problems of randomness. It serves as some motivation for the dryer stuff to follow.

If we toss a coin 100 times and it shows each time Head, we feel lifted to a land of wonders, like Rosencrantz and Guildenstern in [48]. The argument that 100 heads are just as probable as any other outcome convinces us only that the axioms of Probability Theory, as developed in [27], do not solve all mysteries they are sometimes supposed to. We feel that the sequence consisting of 100 heads

is *not random*, though others with the same probability are. Due to the somewhat philosophical character of this paradox, its history is marked by an amount of controversy unusual for ordinary mathematics. Before the reader hastes to propose a solution, let us consider a less trivial example, due to L. A. Levin.

Suppose that in some country, the share of votes for the ruling party in 30 consecutive elections formed a sequence $0.99x_i$ where for every even i, the number x_i is the i-th digit of $\pi = 3.1415\ldots$. Though many of us would feel that the election results were manipulated, it turns out to be surprisingly difficult to prove this by referring to some general principle.

In a sequence of n fair elections, every sequence ω of n digits has approximately the probability $Q_n(\omega) = 10^{-n}$ to appear as the actual sequence of third digits. Let us fix n. We are given a particular sequence ω and want to *test* the validity of the government's claim that the elections were fair. We interpret the assertion "ω is random with respect to Q_n" as a synonym for "there is no reason to challenge the claim that ω arose from the distribution Q_n".

How can such a claim be challenged at all? The government, just like the weather forecaster who announces 30% chance of rain, does not guarantee any particular set of outcomes. However, to stand behind its claim, it must agree to any *bet* based on the announced distribution. Let us call a *payoff function* with respect the distribution P any nonnegative function $t(\omega)$ with $\sum_\omega P(\omega)t(\omega) \le 1$. If a "nonprofit" gambling casino asks 1 dollar for a game and claims that each outcome has probability $P(\omega)$ then it must agree to pay $t(\omega)$ dollars on outcome ω. We would propose to the government the following payoff function t_0 with respect to Q_n: let $t_0(\omega) = 10^{n/2}$ for all sequences ω whose even digits are given by π, and 0 otherwise. This bet would cost the government $10^{n/2} - 1$ dollars.

Unfortunately, we must propose the bet *before* the elections take place and it is unlikely that we would have come up exactly with the payoff function t_0. Is then the suspicion unjustifiable?

No. Though the function t_0 is not as natural as to guess it in advance, it is still highly "regular". And already Laplace assumed in [15] that the number of "regular" bets is so small we can afford to make them *all* in advance and still win by a wide margin.

Kolmogorov discovered in [26] and [28] (probably without knowing about [15] but following a four decades long controversy on von Mises' concept of randomness, see [51]) that to make this approach work we must define "regular" or "simple" as "having a short description" (in some formal sense to be specified below). There cannot be many objects having a short description because there are not many short strings of symbols to be used as descriptions.

We thus come to the principle saying that on a random outcome, all sufficiently simple payoff functions must take small values. It turns out below that this

can be replaced by the more elegant principle saying that *a random outcome* itself *is not too simple*. If descriptions are written in a 2-letter alphabet then a typical sequence of n digits takes $n \log 10$ letters to describe (if not stated otherwise, all logarithms in these notes are to the base 2). The digits of π can be generated by an algorithm whose description takes up only some constant length. Therefore the sequence $x_1 \ldots x_n$ above can be described with approximately $(n/2) \log 10$ letters, since every other digit comes from π. It is thus significantly simpler than a typical sequence and can be declared nonrandom.

1.1.1 Formal results

The theory of randomness is more impressive for infinite sequences than for finite ones, since sharp distinction can be made between random and nonrandom infinite sequences. For technical simplicity, first we will confine ourselves to finite sequences, especially a *discrete sample space* Ω, which we identify with the set of natural numbers. Binary strings will be identified with the natural numbers they denote.

Definition 1.1.1 A Turing machine is an imaginary computing device consisting of the following. A *control state* belonging to a finite set A of possible control states. A fixed number of infinite (or infinitely extendable) strings of cells called *tapes*. Each cell contains a symbol belonging to a finite *tape alphabet* B. On each tape, there is a read-write head observing one of the tape cells. The machine's *configuration* (global state) is determined at each instant by giving its control state, the contents of the tapes and the positions of the read-write heads. The "hardware program" of the machine determines its configuration in the next step as a functon of the control state and the tape symbols observed. It can change the control state, the content of the observed cells and the position of the read-write heads (the latter by one step to the left or right). Turing machines can emulate computers of any known design (see for example [58]). The tapes are used for storing programs, input data, output and as memory. ⌟

The Turing machines that we will use to interpret descriptions will be somewhat restricted.

Definition 1.1.2 Consider a Turing machine F which from a binary string p and a natural number x computes the output $F(p, x)$ (if anything at all). We will say that F *interprets* p as a *description* of $F(p, x)$ in the presence of the side information x. We suppose that if $F(p, x)$ is defined then $F(q, x)$ is not defined for any prefix q of p. Such machines are called *self-delimiting* (s.d.).

The *conditional complexity* $H_F(x \mid y)$ of the number x with respect to the number y is the length of the shortest description p for which $F(p, y) = x$. ⌟

Kolmogorov and Solomonoff observed that the function $H_F(x \mid y)$ depends only weakly on the machine F, because there are universal Turing machines capable of simulating the work of any other Turing machine whose description is supplied. More formally, the following theorem holds:

Theorem 1.1.1 (Invariance Theorem) *There is a s.d. Turnig machine T such that for any s.d. machine F a constant c_F exists such that for all x, y we have $H_T(x \mid y) \leq H_F(x \mid y) + c_F$.*

This theorem motivates the following definition.

Definition 1.1.3 Let us fix T and define $H(x \mid y) = H_T(x \mid y)$ and $H(x) = H(x \mid 0)$. ⌟

The function $H(x \mid y)$ is not computable. We can compute a nonincreasing, convergent sequence of approximations to $H(x)$ (it is *semicomputable* from above), but will not know how far to go in this sequence for some prescribed accuracy.

If x is a binary string of length n then $H(x) \leq n + 2\log n + c$ for some constant c. The description of x with this length gives x bit-for-bit, along with some information of length $2\log n$ which helps the s.d. machine find the end of the description. For most binary strings of lenght n, no significantly shorter description exists, since the number of short descriptions is small. Below, this example is generalized and sharpened for the case when instead of counting the number of simple sequences, we measure their probability.

Definition 1.1.4 Denote by x^* the first one among the shortest descriptions of x. ⌟

The correspondence $x \to x^*$ is a *code* in which no codeword is the prefix of another one. This implies by an argument well-known in Information Theory the inequality

$$\sum_x 2^{-H(x|y)} \leq 1, \tag{1.1.1}$$

hence only a few objects x can have small complexity. The converse of the same argument goes as follows. Let μ be a *computable* probability distribution, one for which there is a binary program computing $\mu(\omega)$ for each ω to any given degree of accuracy. Let $H(\mu)$ be the length of the shortest one of these programs. Then

$$H(\omega) \leq -\log \mu(\omega) + H(\mu) + c. \tag{1.1.2}$$

Here c is a universal constant. These two inequalities are the key to the estimates of complexity and the characterization of randomness by complexity.

Denote

$$d_\mu(\omega) = -\log \mu(\omega) - H(\omega).$$

Inequality (1.1.1) implies

$$t_\mu(\omega) = 2^{d_\mu(\omega)}$$

can be viewed as a *payoff function*. Now we are in a position to solve the election paradox. We propose the payoff function

$$2^{-\log Q_n(\omega) - H(\omega|n)}$$

to the government. (We used the conditional complexity $H(\omega \mid n)$ because the uniform distribution Q_n depends on n.) If every other digit of the outcome x comes from π then $H(x \mid n) \le (n/2)\log 10 + c_0$ hence we win a sum $2^{t(x)} \ge c_1 10^{n/2}$ from the government (for some constants $c_0, c_1 > 0$), even though the bet does not contain any reference to the number π.

The fact that $t_\mu(\omega)$ is a payoff function implies by Markov's Inequality for any $k > 0$

$$\mu\{\omega : H(\omega) < -\log \mu(\omega) - k\} < 2^{-k}. \tag{1.1.3}$$

Inequalities (1.1.2) and (1.1.3) say that with large probability, the complexity $H(\omega)$ of a random outcome ω is close to its upper bound $-\log \mu(\omega) + H(\mu)$. This law occupies distinguished place among the "laws of probability", because if the outcome ω violates *any* such law, the complexity falls far below the upper bound. Indeed, a proof of some "law of probability" (like the law of large numbers, the law of iterated logarithm, etc.) always gives rise to some simple computable payoff function $t(\omega)$ taking large values on the outcomes violating the law, just as in the election example. Let m be some large number, suppose that $t(\omega)$ has complexity $< m/2$, and that $t(\omega_0) > 2^m$. Then inequality (1.1.2) can be applied to $\nu(\omega) = \mu(\omega)t(\omega)$, and we get

$$\begin{aligned} H(\omega) &\le -\log \mu(\omega) - m + H(\nu) + c_0 \\ &\le -\log \mu(\omega) - m/2 + H(\mu) + c_1 \end{aligned}$$

for some constants c_0, c_1.

More generally, the payoff function $t_\mu(\omega)$ is *maximal* (up to a multiplicative constant) among all payoff functions that are semicomputable (from below). Hence the quantity $-\log \mu(\omega) - H(\omega)$ is a *universal test of randomness*. Its value measures the *deficiency of randomness* in the outcome ω with respect to the distribution μ, or the extent of justified suspicion against the hypothesis μ given the outcome ω.

1.1.2 Applications

Algorithmic Information Theory (AIT) justifies the intuition of random sequences as nonstandard analysis justifies infinitely small quantities. Any statement of

classical probability theory is provable without the notion of randomness, but some of them are easier to find using this notion. Due to the incomputability of the universal randomness test, only its approximations can be used in practice.

Pseudorandom sequences are sequences generated by some algorithm, with *some* randomness properties with respect to the coin-tossing distribution. They have very low complexity (depending on the strength of the tests they withstand, see for example [14]), hence are not random. Useful pseudorandom sequences can be defined using the notion of *computational complexity*, for example the number of steps needed by a Turing machine to compute a certain function. The existence of such sequences can be proved using some difficult unproven (but plausible) assumptions of computation theory. See [6], [57], [24], [33].

Inductive inference

The incomputable "distribution" $\mathbf{m}(\omega) = 2^{-H(\omega)}$ has the remarkable property that, the test $d(\omega \mid \mathbf{m})$, shows all outcomes ω "random" with respect to it. Relations (1.1.2) and (1.1.3) can be read as saying that if the real distribution is μ then $\mu(\omega)$ and $\mathbf{m}(\omega)$ are close to each other with large probability. Therefore if we know that ω comes from some unknown simple distribution μ then we can use $\mathbf{m}(\omega)$ as an estimate of $\mu(\omega)$. This suggests to call $\mathbf{m}$ the "apriori probability" (but we will not use this term much). The randomness test $d_\mu(\omega)$ can be interpreted in the framework of hypothesis testing: it is the likelihood ratio between the hypothesis μ and the fixed alternative hypothesis $\mathbf{m}$.

In ordinary statistical hypothesis testing, some properties of the unknown distribution μ are taken for granted. The sample ω is generally a large independent sample: μ is supposed to be a product distribution. Under these conditions, the universal test could probably be reduced to some of the tests used in statistical practice. However, these conditions do not hold in other applications: for example testing for independence in a proposed random sequence, predicting some time series of economics, or pattern recognition.

If the apriori probability $\mathbf{m}$ is a good estimate of the actual probability then we can use the conditional apriori probability for prediction, without reference to the unknown distribution μ. For this purpose, one first has to define apriori probability for the set of infinite sequences of natural numbers, as done in [59]. We denote this function by M. For any finite sequence x, the number $M(x)$ is the apriori probability that the outcome is some extension of x. Let x, y be finite sequences. The formula

$$\frac{M(xy)}{M(x)} \tag{1.1.4}$$

is an estimate of the conditional probability that the next terms of the outcome will be given by y provided that the first terms are given by x. It converges to the actual conditional probability $\mu(xy)/\mu(x)$ with μ-probability 1 for any computable distribution μ (see for example [45]). To understand the surprising generality of the formula (1.1.4), suppose that some infinite sequence $z(i)$ is given in the following way. Its even terms are the subsequent digits of π, its odd terms are uniformly distributed, independently drawn random digits. Let

$$z(1:n) = z(1)\cdots z(n).$$

Then $M(z(1:2i)a/M(z(1:2i))$ converges to 0.1 for $a = 0, \ldots, 9$, while $M(z(1:2i+1)a)/M(z(1:2i+1))$ converges to 1 if a is the i-th digit of π, and to 0 otherwise.

The inductive inference formula using conditional apriori probability can be viewed as a mathematical form of "Occam's Razor": the advice to predict by the simplest rule fitting the data. It can also viewed as a realization of Bayes'Rule, with a universally applicable apriori distribution. Since the distribution M is incomputable, we view the main open problem of inductive inference to find maximally efficient approximations to it. Sometimes, even a simple approximation gives nontrivial results (see [3]).

Information theory

Since with large probability, $H(\omega)$ is close to $-\log\mu(\omega)$, the *entropy* $-\sum_\omega \mu(\omega)\log\mu(\omega)$ of the distribution μ is close to the *average complexity* $\sum_\omega \mu(\omega)H(\omega)$. The complexity $H(x)$ of an object x can indeed be interpreted as the distribution-free definition of *information content*. The correspondence $x \mapsto x^*$ is a sort of *universal code*: its average (even individual) "rate", or codeword length is almost equally good for any simple computable distribution.

It is of great conceptual help to students of statistical physics that entropy can be defined now not only for ensembles (probability distributions), but for individual states of a physical system. The notion of an individual *incompressible sequence*, a sequence whose complexity is maximal, proved also extremely useful in finding information-theoretic lower bounds on the computing speed of certain Turing machines (see [38]).

The conditional complexity $H(x \mid y)$ obeys identities analogous to the information-theoretical identities for conditional entropy, but these identities are less trivial to prove in AIT. The information $I(x:y) = H(x) + H(y) - H(x,y)$ has several interpretations. It is known that $I(x:y)$ is equal, to within an additive constant, to $H(y) - H(y \mid x^*)$, the amount by which the object x^* (as defined in the previous section) decreases our uncertainty about y. But it can be written as

$-\log \mathbf{m}^2(x,y) - H(x,y) = d((x,y) \mid \mathbf{m}^2)$ where $\mathbf{m}^2 = \mathbf{m} \times \mathbf{m}$ is the product distribution of $\mathbf{m}$ with itself. It is thus the deficiency of randomness of the pair (x,y) with respect to this product distribution. Since any object is random with respect to $\mathbf{m}$, we can view the randomness of the pair (x,y) with respect to the product $\mathbf{m}^2$ as the *independence* of x and y from each other. Thus "information" measures the "deficiency of independence".

Logic

Some theorems of Mathematical Logic (in particular, Gödel's theorem) have a strong quantitative form in AIT, with new philosophical implications (see [8], [9], [30], [32]). Levin based a new system of intuitionistic analysis on his Independence Principle (see below) in [32].

1.1.3 History of the problem

P. S. Laplace thought that the number of "regular" sequences (whatever "regular" means) is much smaller than the number of irregular ones (see [15]). In the first attempt at formalization hundred years later, R. von Mises defined an infinite binary sequence as random (a "Kollektiv") if the relative frequencies converge in any subsequence selected according to some (non-anticipating) "rule" (whatever "rule" means, see [51]). As pointed out by A. Wald and others, Mises's definitions are sound only if a countable set of possible rules is fixed. The logician A. Church, in accordance with his famous thesis, proposed to understand "rule" here as "recursive (computable) function".

The Mises selection rules can be considered as special randomness tests. In the ingenious work [50], J. Ville proved that they do not capture all relevant properties of random sequences. In particular, a Kollektiv can violate the law of iterated logarithm. He proposed to consider arbitrary payoff functions (a countable set of them), as defined on the set of infinite sequences—these are more commonly known as *martingales*.

For the solution of the problem of inductive inference, R. Solomonoff introduced complexity and apriori probability in [44] and proved the Invariance Theorem. A. N. Kolmogorov independently introduced complexity as a measure of individual information content and randomness, and proved the Invariance Theorem (see [26] and [28]). The incomputability properties of complexity have noteworthy philosophical implications (see [4], [8], [9].

P. Martin-Löf defined in [37] randomness for infinite sequences. His concept is essentially the synthesis of Ville and Church (as noticed in [40]). He recognized

the existence of a universal test, and pointed out the close connection between the randomness of an infinite sequence and the complexity of its initial segments.

L. A. Levin defined the apriori probability M as a maximal (to within a multiplicative constant) semicomputable measure. With the help of a modified complexity definition, he gave a simple and general characterization of random sequences by the behavior of the complexity of their initial segments (see [59], [29]). In [17] and [30], the information-theoretical properties of the self-delimiting complexity (as defined above) are exactly described. See also [10], [41] and [56].

In [32], Levin defined a deficiency of randomness $d(\omega \mid \mu)$ in a uniform manner for all (computable or incomputable) measures μ. He proved that all outcomes are random with respect to the apriori probability M. In this and earlier papers, he also proved the Law of Information Conservation, stating that the information $I(\alpha;\beta)$ in a sequence α about a sequence β cannot be significantly increased by any algorithmic processing of α (even using random number generators). He derived this law from a so-called Law of Randomness Conservation via the definition of information $I(\alpha;\beta)$ as deficiency of randomness with respect to the product distribution M^2. Levin suggested the Independence Principle saying that any sequence α arising in *nature* contains only finite information $I(\alpha;\beta)$ about any sequence β defined by *mathematical* means. With this principle, he showed that the use of more powerful notions of definability in randomness tests (or the notion of complexity) does not lead to fewer random sequences among those arising in nature.

The monograph [16] is very useful as background information on the various attempts in this century at solving the paradoxes of probability theory. The work [32] is quite comprehensive but very dense; it can be recommended only to devoted readers. The work [59] is comprehensive and readable but not quite up-to-date. The surveys [12], [40] and [42] can be used to complement it.

The most up-to-date and complete survey, which subsumes most of these notes, is [35].

AIT created many interesting problems of its own. See for example [10], [11], [17], [18], [19], [20], [34], [32], [36], [40], [43], [46], and the technically difficult results [47] and [53].

Acknowledgment

The form of exposition of the results in these notes and the general point of view represented in them were greatly influenced by Leonid Levin. More recent communication with Paul Vitányi, Mathieu Hoyrup, Cristóbal Rojas and Alexander Shen has also been very important.

1.2 Notation

When not stated otherwise, log means base 2 logarithm. The cardinality of a set A will be denoted by $|A|$. (Occasionally there will be inconsistency, sometimes denoting it by $|A|$, sometimes by $\#A$.) If A is a set then $1_A(x)$ is its indicator function:

$$1_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{otherwise.} \end{cases}$$

The empty string is denoted by Λ. The set A^* is the set of all finite strings of elements of A, including the empty string. Let $l(x)$ denote the length of string x. (Occasionally there will be inconsistency, sometimes denoting it by $|x|$, sometimes by $l(x)$). For sequences x and y, let $x \sqsubseteq y$ denote that x is a prefix of y. For a string x and a (finite or infinite) sequence y, we denote their concatenation by xy. For a sequence x and $n \le l(x)$, the n-th element of x is $x(n)$, and

$$x(i : j) = x(i) \cdots x(j).$$

Sometimes we will also write

$$x^{\le n} = x(1 : n).$$

The string $x_1 \cdots x_n$ will sometimes be written also as $(x_1, \ldots, x_n)$. For natural number m let $\beta(m)$ be the binary sequence denoting m in the binary notation. We denote by $X^{\mathbb{N}}$ the set of all infinite sequences of elements of X.

The sets of natural numbers, integers, rational numbers, real numbers and complex numbers will be denoted respectively by $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$. The set of non-negative real numbers will be denoted by $\mathbb{R}_+$. The set of real numbers with $-\infty$, ∞ added (with the appropriate topology making it compact) will be denoted by $\overline{\mathbb{R}}$. Let

$$\mathbb{S}_r = \{0, \ldots, r-1\}^*, \quad \mathbb{S} = \mathbb{N}^*, \quad \mathbb{B} = \{0, 1\}.$$

We use $\wedge$ and $\vee$ to denote min and max, further

$$|x|^+ = x \vee 0, \quad |x|^- = |-x|^+$$

for real numbers x.

Let $\langle\cdot\rangle$ be some standard one-to-one encoding of $\mathbb{N}^*$ to $\mathbb{N}$, with partial inverses $[\cdot]_i$ where $[\langle x\rangle]_i = x(i)$ for $i \le l(x)$. For example, we can have

$$\langle i, j\rangle = \frac{1}{2}(i+1)(i+j+1) + j, \quad \langle n_1, \ldots, n_{k+1}\rangle = \langle\langle n_1, \ldots, n_k\rangle, n_{k+1}\rangle.$$

We will use $\langle \cdot, \cdot \rangle$ in particular as a pairing function over $\mathbb{N}$. Similar pairing functions will be assumed over other domains, and they will also be denoted by $\langle \cdot, \cdot \rangle$.

Another use of the notation $(\cdot, \cdot)$ may arise when the usual notation (x, y) of an ordered pair of real numbers could be confused with the same notation of an open interval. In this case, we will use (x, y) to denote the pair.

The relations

$$f \stackrel{+}{<} g, \quad f \stackrel{*}{<} g$$

mean inequality to within an additive constant and multiplicative constant respectively. The first is equivalent to $f \leq g + O(1)$, the second to $f = O(g)$. The relation $f \stackrel{*}{=} g$ means $f \stackrel{*}{<} g$ and $f \stackrel{*}{>} g$.

1.3 Kolmogorov complexity

1.3.1 Invariance

It is natural to try to define the *information content* of some text as the size of the smallest string (code) from which it can be reproduced by some decoder, interpreter. We do not want too much information "to be hidden in the decoder" we want it to be a "simple" function. Therefore, unless stated otherwise, we require that our interpreters be *computable*, that is partial recursive functions.

Definition 1.3.1 A partial recursive function A from $\mathbb{S}_2 \times \mathbb{S}$ to $\mathbb{S}$ will be called a (binary) *interpreter*, or *decoder*. We will often refer to its first argument as the *program*, or *code*. ⌟

Partial recursive functions are relatively simple; on the other hand, the class of partial recursive functions has some convenient closure properties.

Definition 1.3.2 For any binary interpreter A and strings $x, y \in \mathbb{S}$, the number

$$K_A(x \mid y) = \min\{l(p) : A(p, y) = x\}$$

is called the *conditional Kolmogorov-complexity of x given y, with respect to the interpreter A* . (If the set after the "min" is empty, then the minimum is ∞). We write $K_A(x) = K_A(x \mid \Lambda)$. ⌟

The number $K_A(x)$ measures the length of the shortest description for x when the algorithm A is used to interpret the descriptions.

The value $K_A(x)$ depends, of course, on the underlying function A. But, as Theorem 1.3.1 shows, if we restrict ourselves to sufficiently powerful interpreters A, then switching between them can change the complexity function only by amounts bounded by some additive constant. Therefore complexity can be considered an intrinsic characteristic of finite objects.

Definition 1.3.3 A binary p.r. interpreter U is called *optimal* if for any binary p.r. interpreter A there is a constant $c < \infty$ such that for all x, y we have

$$K_U(x \mid y) \leq K_A(x \mid y) + c. \tag{1.3.1}$$

⌟

Theorem 1.3.1 (Invariance Theorem) *There is an optimal p.r. binary interpreter.*

Proof. The idea is to use interpreters that come from universal partial recursive functions. However, not all such functions can be used for universal interpreters. Let us introduce an appropriate pairing function. For $x \in \mathbb{B}^n$, let

$$x^o = x(1)0x(2)0 \ldots x(n-1)0x(n)1$$

where $x^o = x1$ for $l(x) = 1$. Any binary sequence x can be uniquely represented in the form $p = a^o b$.

We know there is a p.r. function $V : \mathbb{S}_2 \times \mathbb{S}_2 \times \mathbb{S} \to \mathbb{S}$ that is *universal*: for any p.r. binary interpreter A, there is a string a such that for all p, x, we have $A(p, x) = V(a, p, x)$. Let us define the function $U(p, x)$ as follows. We represent the string p in the form $p = u^o v$ and define $U(p, x) = V(u, v, x)$. Then U is a p.r. interpreter. Let us verify that it is optimal. Let A be a p.r. binary interpreter, a a binary string such that $A(p, x) = U(a, p, x)$ for all p, x. Let x, y be two strings. If $K_A(x \mid y) = \infty$, then (1.3.1) holds trivially. Otherwise, let p be a binary string of length $K_A(x \mid y)$ with $A(p, y) = x$. Then we have

$$U(a^o p, y) = V(a, p, y) = A(p, y) = x,$$

and

$$K_U(x \mid y) \leq 2l(a) + K_A(x \mid y).$$

□

The constant $2l(a)$ is in the above proof a bound on the complexity of description of the interpreter A for the optimal interpreter U. Let us note that for any two optimal interpreters $U^{(1)}, U^{(2)}$, there is a constant c such that for all x, y, we have

$$|K_{U^{(1)}}(x \mid y) - K_{U^{(2)}}(x \mid y)| < c. \tag{1.3.2}$$

Hence the complexity $K_U(x)$ of description of an object x does not depend strongly on the interpreter U. Still, for every string x, there is an optimal interpreter U with $K_U(x) = 0$. Imposing a universal bound on the table size of the Turing machines used to implement optimal interpreters, we can get a universal bound on the constants in (1.3.2).

The theorem motivates the following definition.

Definition 1.3.4 We fix an optimal binary p.r. interpreter U and write $K(x \mid y)$ for $K_U(x \mid y)$. ⌟

Theorem 1.3.1 (as well as other invariance theorems) is used in AIT for much more than just to show that $K_A(x)$ is a proper concept. It is the principal tool to find upper bounds on $K(x)$; this is why most such upper bounds are proved to hold only to within an additive constant.

The optimal interpreter $U(p, x)$ defined in the proof of Theorem 1.3.1 is obviously a universal partial recursive function. Because of its convenient properties we will use it from now on as our standard universal p.r. function, and we will refer to an arbitrary p.r. function as $U_p(x) = U(p, x)$.

Definition 1.3.5 We define $U_p(x_1, \ldots, x_k) = U_p(\langle x_1 \ \ldots, x_k\rangle)$. Similarly, we will refer to an arbitrary computably enumerable set as the range W_p of some U_p. We often do not need the second argument of the function $U(p, x)$. We therefore define

$$U(p) = U(p, \Lambda).$$

⌟

It is of no consequence that we chose binary strings as as descriptions. It is rather easy to define, (Exercise) for any two natural numbers r, s, a standard encoding cnv_s^r of base r strings x into base s strings with the property

$$l(\mathrm{cnv}_s^r(x)) \leq l(x)\frac{\log r}{\log s} + 1.$$

Now, with r-ary strings as descriptions, we must define $K_A(x)$ as the minimum of $l(p)\log r$ over all programs p in $\mathbb{S}_r$ with $A(p) = x$. The equivalence of the definitions for different values of r is left as an exercise.

1.3.2 Simple quantitative estimates

We found it meaningful to speak about the information content, complexity of a finite object. But how to compute this quantity? It turns out that complexity is not a computable function, but much is known about its statistical behavior.

The following notation will be useful in the future.

Definition 1.3.6 The relation $f \stackrel{+}{<} g$ means inequality to within an additive constant, that there is a constant c such that for all x, $f(x) \leq g(x)$. We can write this also as $f \leq g + O(1)$. We also say that g *additively dominates* f. The relation $f \stackrel{+}{=} g$ means $f \stackrel{+}{<} g$ and $f \stackrel{+}{>} g$. The relation $f \stackrel{*}{<} g$ among nonnegative functions means inequality to within a multiplicative constant. It is equivalent to $\log f \stackrel{+}{<} \log g$ and

$f = O(g)$. We also say that g *multiplicatively dominates* f. The relation $f \stackrel{*}{=} g$ means $f \stackrel{*}{<} g$ and $f \stackrel{*}{>} g$. ⌟

With the above notation, here are the simple properties.

Theorem 1.3.2 *The following simple upper bound holds.*

a) *For any natural number m, we have*

$$K(m) \stackrel{+}{<} \log m. \tag{1.3.3}$$

b) *For any positive real number v and string y, every finite set E of size m has at least $m(1 - 2^{-v+1})$ elements x with $K(x \mid y) \geq \log m - v$.*

Corollary 1.3.7 $\lim_{n\to\infty} K(n) = \infty$.

Theorem 1.3.2 suggests that if there are so few strings of low complexity, then $K(n)$ converges fast to ∞. In fact this convergence is extremely slow, as shown in a later section.

Proof of Theorem 1.3.2. First we prove (a). Let the interpreter A be defined such that if $\beta(m) = p$ then $A(p, y) = m$. Since $|\beta(m)| = \lceil \log m \rceil$, we have $K(m) \stackrel{+}{<} K_A(m) < \log m + 1$.

Part (b) says that the trivial estimate (1.3.3) is quite sharp for *most* numbers under m. The reason for this is that since there are only few short programs, there are only few objects of low complexity. For any string y, and any positive real natural number u, we have

$$|\{x : K(x \mid y) \leq u\}| < 2^{u+1}. \tag{1.3.4}$$

To see this, let $n = \lfloor \log u \rfloor$. The number $2^{n+1} - 1$ of different binary strings of length $\leq n$ is an upper bound on the number of different shortest programs of length $\leq u$. Now (b) follows immediately from (1.3.4). □

The three properties of complexity contained in Theorems 1.3.1 and 1.3.2 are the ones responsible for the applicabiliy of the complexity concept. Several important later results can be considered transformations, generalizations or analogons of these.

Let us elaborate on the upper bound (1.3.3). For any string $x \in \mathbb{S}_r$, we have

$$K(x) \stackrel{+}{<} n \log r + 2 \log r. \tag{1.3.5}$$

In particular, for any binary string x, we have

$$K(x) \stackrel{+}{<} l(x).$$

Indeed, let the interpreter A be defined such that if $p = \beta(r)^o \mathrm{cnv}_2^r(x)$ then $A(p, y) = x$. We have $K_A(x) \le 2^{l(\beta(r))} + n \log r + 1 \le (n+2)\log r + 3$. Since $K(x) \stackrel{+}{<} K_A(x)$, we are done.

We will have many more upper bound proofs of this form, and will not always explicitly write out the reference to $K(x \mid y) \stackrel{+}{<} K_A(x \mid y)$, needed for the last step.

Apart from the conversion, inequality (1.3.5) says that since the beginning of a program can command the optimal interpreter to copy the rest, the complexity of a binary sequence x of length n is not larger than n.

Inequality (1.3.5) contains the term $2\log r$ instead of $\log r$ since we have to apply the encoding w^o to the string $w = \beta(r)$; otherwise the interpreter cannot detach it from the program p. We could use the code $\beta(|w|)^o w$, which is also a prefix code, since the first, detachable part of the codeword tells its length.

For binary strings x, natural numbers n and real numbers $u \ge 1$ we define

$$\begin{aligned} J(u) &= u + 2\log u, \\ \iota(x) &= \beta(|x|)^o x, \\ \iota(n) &= \iota(\beta(n)). \end{aligned}$$

We have $l(\iota(x)) \stackrel{+}{<} J(l(x))$ for binary sequences x, and $l(\iota(r)) \stackrel{+}{<} J(\log r)$ for numbers r. Therefore (1.3.5) is true with $J(\log r)$ in place of $2\log r$. Of course, $J(x)$ could be replaced by still smaller functions, for example $x + J(\log x)$. We return to this topic later.

1.4 Simple properties of information

If we transform a string x by applying to it a p.r. function, then we cannot gain information over the amount contained in x plus the amount needed to describe the transformation. The string x becomes easier to describe, but it helps less in describing other strings y.

Theorem 1.4.1 *For any partial recursive function U_q, over strings, we have*

$$\begin{aligned} K(U_q(x) \mid y) &\stackrel{+}{<} K(x \mid y) + J(l(q)), \\ K(y \mid U_q(x)) &\stackrel{+}{>} K(y \mid x) - J(l(q)). \end{aligned}$$

Proof. To define an interpreter A with $K_A(U_q(x) \mid y) \stackrel{+}{<} K(x \mid y) + J(l(q))$, we define $A(\iota(q)p, y) = U_q(U(p, y))$. To define an interpreter B with $K_B(y \mid x) \stackrel{+}{<} K(y \mid U_q(x)) + J(q)$, we define $B(\iota(q)p, x) = U(p, U_q(x))$. □

The following notation is natural.

Definition 1.4.1 The definition of conditional complexity is extended to pairs, etc. by

$$K(x_1, \ldots, x_m \mid y_1, \ldots, y_n) = K(\langle x_1, \ldots, x_m\rangle \mid \langle y_1, \ldots, y_n\rangle).$$

⌟

With the new notation, here are some new results.

Corollary 1.4.2 *For any one-to-one p.r. function U_p, we have*

$$|K(x) - K(U_p(x))| \stackrel{+}{<} J(l(p)).$$

Further,

$$\begin{aligned}
K(x \mid y, z) &\stackrel{+}{<} K(x \mid U_p(y), z) + J(l(p)),\\
K(U_p(x)) &\stackrel{+}{<} K(x) + J(l(p)),\\
K(x \mid z) &\stackrel{+}{<} K(x, y \mid z), \qquad (1.4.1)\\
K(x \mid y, z) &\stackrel{+}{<} K(x \mid y),\\
K(x, x) &\stackrel{+}{=} K(x), \qquad (1.4.2)\\
K(x, y \mid z) &\stackrel{+}{=} K(y, x \mid z),\\
K(x \mid y, z) &\stackrel{+}{=} K(x \mid z, y),\\
K(x, y \mid x, z) &\stackrel{+}{=} K(y \mid x, z),\\
K(x \mid x, z) &\stackrel{+}{=} K(x \mid x) \stackrel{+}{=} 0.
\end{aligned}$$

We made several times implicit use of a basic additivity property of complexity which makes it possible to estimate the joint complexity of a pair by the complexities of its constituents. As expressed in Theorem 1.4.2, it says essentially that to describe the pair of strings, it is enough to know the description of the first member and of a method to find the second member using our knowledge of the first one.

Theorem 1.4.2

$$K(x, y) \stackrel{+}{<} J(K(x)) + K(y \mid x).$$

Proof. We define an interpreter A as follows. We decompose any binary string p into the form $p = \iota(w)q$, and let $A(p, z) = U(q, U(w))$. Then $K_A(x, y)$ is bounded, to within an additive constant, by the right-hand side of the above inequality. □

Corollary 1.4.3 *For any p.r. function U_p over pairs of strings, we have*

$$\begin{aligned} K(U_p(x,y)) &\stackrel{+}{<} J(K(x)) + K(y \mid x) + J(l(p)) \\ &\stackrel{+}{<} J(K(x)) + K(y) + J(l(p)), \end{aligned}$$

and in particular,

$$K(x,y) \stackrel{+}{<} J(K(x)) + K(y).$$

This corollary implies the following continuity property of the function $K(n)$: for any natural numbers n, h, we have

$$|K(n+h) - K(n)| \stackrel{+}{<} J(\log h). \tag{1.4.3}$$

Indeed, $n+h$ is a recursive function of n and h. The term $2\log K(x)$ making up the difference between $K(x)$ and $J(K(x))$ in Theorem 1.4.2 is attributable to the fact that minimal descriptions cannot be concatenated without loosing an "end-marker". It cannot be eliminated, since there is a constant c such that for all n, there are binary strings x, y of length $\leq n$ with

$$K(x) + K(y) + \log n < K(x,y) + c.$$

Indeed, there are $n2^n$ pairs of binary strings whose sum of lengths is n. Hence by Theorem 1.3.2 b, there will be a pair (x,y) of binary strings whose sum of lengths is n, with $K(x,y) > n + \log n - 1$. For these strings, inequality (1.3.5) implies $K(x) + K(y) \stackrel{+}{<} l(x) + l(y)$. Hence $K(x) + K(y) + \log n \stackrel{+}{<} n + \log n < K(x,y) + 1$.

Regularities in a string will, in general, decrease its complexity radically. If the whole string x of length n is given by some rule, that is we have $x(k) = U_p(k)$, for some recursive function U_p, then

$$K(x) \stackrel{+}{<} K(n) + J(l(p)).$$

Indeed, let us define the p.r. function $V(q,k) = U_q(1)\ldots U_q(k)$. Then $x = V(p,n)$ and the above estimate follows from Corollary 1.4.2.

For another example, let $x = y_1y_1y_2y_2\cdots y_ny_n$, and $y = y_1y_2\cdots y_n$. Then $K(x) \stackrel{+}{=} K(y)$ even though the string x is twice longer. This follows from Corollary 1.4.2 since x and y can be obtained from each other by a simple recursive operation.

Not all "regularities" decrease the complexity of a string, only those which distinguish it from the mass of all strings, putting it into some class which is both small and algorithmically definable. For a binary string of length n, it is nothing unusual to have its number of 0-s between $n/2 - \sqrt{n}$ and $n/2$. Therefore such

strings can have maximal or almost maximal complexity. If x has k zeroes then the inequality

$$K(x) \stackrel{+}{<} \log \binom{n}{k} + J(\log n) + J(\log k)$$

follows from Theorems 1.3.2 and 1.4.3.

Theorem 1.4.3 *Let $E = W_p$ be an enumerable set of pairs of strings defined enumerated with the help of the program p for example as*

$$W_p = \{U(p, x) : x \in \mathbb{N}\}. \tag{1.4.4}$$

We define the section

$$E^a = \{x : \langle a, x\rangle \in E\}. \tag{1.4.5}$$

Then for all a and $x \in E^a$, we have

$$K(x \mid a) \stackrel{+}{<} \log |E^a| + J(l(p)).$$

Proof. We define an interpreter $A(q, b)$ as follows. We decompose q into $q = \iota(p)\beta(t)$. From a standard enumeration $(a(k), x(k))$ $(k = 1, 2, \ldots)$ of the set W_p, we produce an enumeration $x(k, b)$ $(k = 1, 2, \ldots)$, without repetition, of the set W_p^b for each b, and define $A(q, b) = x(t, b)$. For this interpreter A, we get $k_A(x \mid a) \stackrel{+}{<} J(l(p)) + \log |E^a|$. □

It follows from Theorem 1.3.2 that whenever the set E^a is finite, the estimate of Theorem 1.4.3 is sharp for most of its elements x.

The shortest descriptions of a string x seem to carry some extra information in them, above the description of x. This is suggested by the following strange identities.

Theorem 1.4.4 *We have $K(x, K(x)) \stackrel{+}{=} K(x)$.*

Proof. The inequality $\stackrel{+}{>}$ follows from (1.4.1). To prove $\stackrel{+}{<}$, let $A(p) = \langle U(p, l(p))\rangle$. Then $K_A(x, K(x)) \leq K(x)$. □

More generally, we have the following inequality.

Theorem 1.4.5

$$K(y \mid x,\ i - K(y \mid x, i)) \stackrel{+}{<} K(y \mid x, i).$$

The proof is exercise.

The above inequalities are in some sense "pathological", and do not necessarily hold for all "reasonable" definitions of descriptional complexity.

1.5 Algorithmic properties of complexity

The function $K(x)$ is not computable, as it will be shown in this section. However, it has a property closely related to computability. Let $\mathbb{Q}$ be the set of rational numbers.

Definition 1.5.1 (Computability) Let $f : \mathbb{S} \to \mathbb{R}$ be a function. It is *computable* if there is a recursive function $g(x, n)$ with rational values, and $|f(x) - g(x, n)| < 1/n$. ⌟

Definition 1.5.2 (Semicomputability) A function $f : \mathbb{S} \to (-\infty, \infty]$ is *(lower) semicomputable* if the set

$$\{(x, r) : x \in \mathbb{S},\ r \in \mathbb{Q},\ r < f(x)\}$$

is recursively enumerable. A function f is called *upper semicomputable* if $-f$ is lower semicomputable. ⌟

It is easy to show the following:

- A function $f : \mathbb{S} \to \mathbb{R}$ is lower semicomputable iff there exists a recursive function with rational values, (or, equivalently, a computable real function) $g(x, n)$ nondecreasing in n, with $f(x) = \lim_{n\to\infty} g(x, n)$.
- A function $f : \mathbb{S} \to \mathbb{R}$ is computable if it is both lower and upper semicomputable.

The notion of semicomputibility is naturally extendable over other discrete domains, like $\mathbb{S}\times\mathbb{S}$. It is also useful to extend it to functions $\mathbb{R} \to \mathbb{R}$. Let $\mathcal{I}$ denote the set of open rational intervals of $\mathbb{R}$, that is

$$\beta = \{(p, q) : p, q \in \mathbb{Q}, p < q\}.$$

Definition 1.5.3 (Computability for real functions) Let $f : \mathbb{R} \to \mathbb{R}$ be a function. It is *computable* if there is a recursively enumerable set $\mathcal{F} \subseteq \beta^2$ such that denoting $\mathcal{F}_J = \{I : (I, J) \in \mathcal{F}\}$ we have

$$f^{-1}(J) = \bigcup_{I\in\mathcal{F}_J} I.$$

⌟

This says that if $f(x) \in I$ then sooner or later we will find an interval $I \in \mathcal{F}_J$ with the property that $f(z) \in J$ for all $z \in I$. Note that computability implies continuity.

Definition 1.5.4 We say that $f : \mathbb{R} \to \overline{\mathbb{R}}$ is *lower semicomputable* if there is a recursively enumerable set $\mathcal{G} \subseteq \mathcal{Q} \times \beta$ such that denoting $\mathcal{G}_q = \{I : (I, q) \in \mathcal{G}\}$ we have

$$f^{-1}((q, \infty)) = \bigcup_{I \in \mathcal{F}_J} I.$$

⌟

This says that if $f(x) > r$ then sooner or later we will find an interval $I \in \mathcal{F}_J$ with the property that $f(z) > q$ for all $z \in I$.

The following facts are useful and simple to verify:

Proposition 1.5.5

a) *The function* $\lceil \cdot \rceil : \mathbb{R} \to \mathbb{R}$ *is lower semicomputable.*

b) *If* $f, g : \mathbb{R} \to \mathbb{R}$ *are computable then their composition* $f(g(\cdot))$ *is, too.*

c) *If* $f : \mathbb{R} \to \mathbb{R}$ *is lower semicomputable and* $g : \mathbb{R} \to \mathbb{R}$ *(or* $\mathbb{S} \to \mathbb{R}$*) is computable then* $f(g(\cdot))$ *is lower semicomputable.*

d) *If* $f : \mathbb{R} \to \mathbb{R}$ *is lower semicomputable and monotonic, and* $g : \mathbb{R} \to \mathbb{R}$ *(or* $\mathbb{S} \to \mathbb{R}$*) is lower semicomputable then* $f(g(\cdot))$ *is lower semicomputable.*

Definition 1.5.6 We introduce a *universal lower semicomputable function*. For every binary string p, and $x \in \mathbb{S}$, let

$$S_p(x) = \sup\{y/z : y, z \in Z, z \neq 0, \langle\langle x \rangle, y, z\rangle \in W_p\}$$

where W_p was defined in (1.4.4). ⌟

The function $S_p(x)$ is lower semicomputable as a function of the pair (p, x), and for different values of p, it enumerates all semicomputable functions of x.

Definition 1.5.7 Let $S_p(x_1, \ldots, x_k) = S_p(\langle x_1, \ldots, x_k\rangle)$. For any lower semicomputable function f, we call any binary string p with $S_p = f$ a *Gödel number* of f. There is no universal computable function, but for any computable function g, we call any number $\langle\langle p\rangle, \langle q\rangle\rangle$ a Gödel number of g if $g = S_p = -S_q$. ⌟

With this notion, we claim the following.

Theorem 1.5.1 *The function* $K(x \mid y)$ *is upper semicomputable.*

Proof. By Theorem 1.3.2 there is a constant c such that for any strings x, y, we have $K(x \mid y) < \log\langle x\rangle + c$. Let some Turing machine compute our optimal binary p.r. interpreter. Let $U^t(p, y)$ be defined as $U(p, y)$ if this machine, when started on input (p, y), gives an output in t steps, undefined otherwise. Let $K^t(x \mid y)$ be the smaller of $\log\langle x\rangle + c$ and

$$\min\{l(p) \leq t : U^t(p, y) = x\}.$$

Then the function $K^t(x \mid y)$ is computable, monotonic in t and $\lim_{t\to\infty} K^t(x \mid y) = K(x \mid y)$. □

The function $K(n)$ is not computable. Moreover, it has no nontrivial partial recursive lower bound.

Theorem 1.5.2 *Let $U_p(n)$ be a partial recursive function whose values are numbers smaller than $K(n)$ whenever defined. Then*

$$U_p(n) \stackrel{+}{<} J(l(p)).$$

Proof. The proof of this theorem resembles "Berry's paradox ", which says: "The least number that cannot be defined in less than 100 words" is a definition of that number in 12 words. The paradox can be used to prove, whenever we agree on a formal notion of "definition", that the sentence in quotes is not a definition.

Let $x(p, k)$ for $k = 1, 2, \ldots$ be an enumeration of the domain of U_p. For any natural number m in the range of U_p, let $f(p, m)$ be equal to the first $x(p, k)$ with $U_p(x(p, k)) = m$. Then by definition, $m < K(f(p, m))$. On the other hand, $f(p, m)$ is a p.r. function, hence applying Corollary 1.4.3 and (1.3.5) we get

$$\begin{aligned} m < K(f(p, m)) &\stackrel{+}{<} K(p) + J(K(m)) \\ &\stackrel{+}{<} l(p) + 1.5 \log m. \end{aligned}$$

□

Church's first example of an undecidable recursively enumerable set used universal partial recursive functions and the diagonal technique of Gödel and Cantor, which is in close connection to Russel's paradox. The set he constructed is *complete* in the sense of "many-one reducibility".

The first example of sets not complete in this sense were Post's simple sets.

Definition 1.5.8 A computably enumerable set is *simple* if its complement is infinite but does not contain any infinite computably enumerable subsets. ⌟

The paradox used in the above proof is not equivalent in any trivial way to Russell's paradox, since the undecidable computably enumerable sets we get from Theorem 1.5.2 are simple. Let $f(n) \leq \log n$ be any recursive function with $\lim_{n\to\infty} f(n) = \infty$. We show that the set $E = \{n : K(n) \leq f(n)\}$ is simple. It follows from Theorem 1.5.1 that E is recursively enumerable, and from Theorem 1.3.2 that its complement is infinite. Let A be any computably enumerable set disjoint from E. The restriction $f_A(n)$ of the function $f(n)$ to A is a p.r. lower bound for $K(n)$. It follows from Theorem 1.5.2 that $f(A)$ is bounded. Since $f(n) \to \infty$, this is possible only if A is finite.

Gödel used the diagonal technique to prove the incompleteness of any sufficiently rich logical theory with a recursively enumerable axiom system. His technique provides for any sufficiently rich computably enumerable theory a concrete example of an undecidable sentence. Theorem 1.5.2 leads to a new proof of this result, with essentially different examples of undecidable propositions. Let us consider any first-order language L containing the language of standard first-order arithmetic. Let $\langle\cdot\rangle$ be a standard encoding of the formulas of this theory into natural numbers. Let the computably enumerable theory T_p be the theory for which the set of codes $\langle\Phi\rangle$ of its axioms Φ is the computably enumerable set W_p of natural numbers.

Corollary 1.5.9 *There is a constant $c < \infty$ such that for any p, if any sentences with the meaning "$m < K(n)$" for $m > J(l(p)) + c$ are provable in theory T_p then some of these sentences are false.*

Proof. For some p, suppose that all sentences "$m < K(n)$" provable in T_p are true. For a sentence Φ, let $T_p \vdash \Phi$ denote that Φ is a theorem of the theory T_p. Let us define the function

$$A(p, n) = \max\{m : T_p \vdash \text{“}m < K(n)\text{”}\}.$$

This function is semicomputable since the set $\{(p, \langle\Phi\rangle) : T_p \vdash \Phi\}$ is recursively enumerable. Therefore there is a binary string a such that $A(p, n) = S(a, \langle p, n\rangle)$. It is easy to see that we have a constant string b such that

$$S(a, \langle p, n\rangle) = S(q, n)$$

where $q = \overline{b}\overline{a}p$. (Formally, this statement follows for example from the so-called S^n_m-theorem.) The function $S_q(n)$ is a lower bound on the complexity function $K(n)$. By an immediate generalization of Theorem 1.5.2 for semicomputable functions), we have $S(q, n) \stackrel{+}{<} J(l(q)) \stackrel{+}{<} J(l(p))$. □

We have seen that the complexity function $K(x \mid y)$ is not computable, and does not have any nontrivial recursive lower bounds. It has many interesting upper bounds, however, for example the ones in Theorems 1.3.2 and 1.4.3. The following theorem characterizes all upper semicomputable upper bounds (*majorants*) of $K(x \mid y)$.

Theorem 1.5.3 (Levin) *Let $F(x, y)$ be a function of strings semicomputable from above. The relation $K(x \mid y) \stackrel{+}{<} F(x, y)$ holds for all x, y if and only if we have*

$$\log |\{x : F(x, y) < m\}| \stackrel{+}{<} m \tag{1.5.1}$$

for all strings y and natural numbers m.

Proof. Suppose that $K(x \mid y) \stackrel{+}{<} F(x,y)$ holds. Then (1.5.1) follows from b of Theorem 1.3.2.

Now suppose that (1.5.1) holds for all y, m. Then let E be the computably enumerable set of triples (x,y,m) with $F(x,y) < m$. It follows from (1.5.1) and Theorem 1.4.3 that for all $(x,y,m) \in E$ we have $K(x \mid y,m) \stackrel{+}{<} m$. By Theorem 1.4.5 then $K(x \mid y) \stackrel{+}{<} m$. □

The minimum of any finite number of majorants is again a majorant. Thus, we can combine different heuristics for recognizing patterns of sequences.

1.6 The coding theorem

1.6.1 Self-delimiting complexity

Some theorems on the addition of complexity do not have as simple a form as desirable. For example, Theorem 1.4.2 implies

$$K(x,y) \stackrel{+}{<} J(K(x)) + K(y).$$

We would like to see just $K(x)$ on the right-hand side, but the inequality $K(x,y) \stackrel{+}{<} K(x)+K(y)$ does not always hold (see Exercise 5). The problem is that if we compose a program for (x,y) from those of x and y then we have to separate them from each other somehow. In this section, we introduce a variant of Kolmogorov's complexity discovered by Levin and independently by Chaitin and Schnorr, which has the property that "programs" are *self-delimiting*: this will free us of the necessity to separate them from each other.

Definition 1.6.1 A set of strings is called *prefix-free* if for any pair x, y of elements in it, x is not a prefix of y.

A one-to-one function into a set of strings is a *prefix code*, or *instantaneous code* if its domain of definition is prefix free.

An interpreter $f(p,x)$ is *self-delimiting (s.d.)* if for each x, the set D_x of strings p for which $f(p,x)$ is defined is a prefix-free set. ⌟

Example 1.6.2 The mapping $p \to p^o$ is a prefix code. ⌟

A self-delimiting p.r. function $f(p)$ can be imagined as a function computable on a special self-delimiting Turing machine.

Definition 1.6.3 A Turing machine $\mathcal{T}$ is *self-delimiting* if it has no input tape, but can ask any time for a new input symbol. After some computation and a few such requests (if ever) the machine decides to write the output and stop. ⌟

The essential difference between a self-delimiting machine $\mathcal{T}$ and an ordinary Turing machine is that $\mathcal{T}$ does not know in advance how many input symbols suffice; she must compute this information from the input symbols themselves, without the help of an "endmarker".

Requiring an interpreter $A(p, x)$ to be self-delimiting in p has many advantages. First of all, when we concatenate descriptions, the interpreter will be able to separate them without endmarkers. Lost endmarkers were responsible for the additional logarithmic terms and the use of the function $J(n)$ in Theorem 1.4.2 and several other formulas for K.

Definition 1.6.4 A self-delimiting partial recursive interpreter T is called *optimal* if for any other s.d. p.r. interpreter F, we have

$$K_T \stackrel{+}{<} K_F. \tag{1.6.1}$$

⌟

Theorem 1.6.1 *Let us prove that there is an optimal s.d. interpreter.*

Proof. The proof of this theorem is similar to the proof of Theorem 1.3.1. We take the universal partial recursive function $V(a, p, x)$. We transform each function $V_a(p, x) = V(a, p, x)$ into a self-delimiting function $W_a(p, x) = W(a, p, x)$, but so as not to change the functions V_a which are already self-delimiting. Then we form T from W just as we formed U from V. It is easy to check that the function T thus defined has the desired properties. Therefore we are done if we construct W.

Let some Turing machine $\mathcal{M}$ compute $y = V_a(p, x)$ in t steps. We define $W_a(p, x) = y$ if $l(p) \le t$ and if $\mathcal{M}$ does not compute in t or fewer steps any output $V_a(q, x)$ from any extension or prefix q of length $\le t$ of p. If V_a is self-delimiting then $W_a = V_a$. But W_a is always self-delimiting. Indeed, suppose that $p_0 \sqsubseteq p_1$ and that $\mathcal{M}$ computes $V_a(p_i, x)$ in t_i steps. The value $W_a(p_i, x)$ can be defined only if $l(p_i) \le t_i$. The definition of W_a guarantees that if $t_0 \le t_1$ then $W_a(p_1, x)$ is undefined, otherwise $W_a(p_0, x)$ is undefined. □

Definition 1.6.5 We fix an optimal self-delimiting p.r. interpreter $T(p, x)$ and write

$$H(x \mid y) = K_T(x \mid y).$$

We call $H(y \mid x)$ the (self-delimiting) *conditional complexity* of x with respect to y. ⌟

We will use the expression "self-delimiting" only if confusion may arise. Otherwise, under complexity, we generally understand the self-delimiting complexity. Let $T(p) = T(p, \Lambda)$. All definitions for unconditional complexity, joint complexity etc. are automatically in force for $H = K_T$.

Let us remark that the s.d. interpreter defined in Theorem 1.6.1 has the following stronger property. For any other s.d. interpreter G there is a string g such that we have

$$G(x) = T(gx) \tag{1.6.2}$$

for all x. We could say that the interpreter T is *universal*.

Let us show that the functions K and H are asymptotically equal.

Theorem 1.6.2

$$K \stackrel{+}{<} H \stackrel{+}{<} J(K). \tag{1.6.3}$$

Proof. Obviously, $K \stackrel{+}{<} H$. We define a self-delimiting p.r. function $F(p, x)$. The machine computing F tries to decompose p into the form $p = u^o v$ such that u is the number $l(v)$ in binary notation. If it succeeds, it outputs $U(v, x)$. We have

$$H(y \mid x) \stackrel{+}{<} K_F(y \mid x) \stackrel{+}{<} K(y \mid x) + 2 \log K(y \mid x).$$

□

Let us review some of the relations proved in Sections 1.3 and 1.4. Instead of the simple estimate $K(x) \stackrel{+}{<} n$ for a binary string x, of length n, only the obvious consequence of Theorem 1.6.2 holds that is

$$H(x) \stackrel{+}{<} J(n).$$

We must similarly change Theorem 1.4.3. We will use the universal p.r. function $T_p(x) = T(p, x)$. The r.e. set V_p is the range of the function T_p. Let $E = V_p$ be an enumerable set of pairs of strings. The section E^a is defined as in (1.4.5). Then for all $x \in E^a$, we have

$$H(x \mid a) \stackrel{+}{<} J(\log |E^a|) + l(p). \tag{1.6.4}$$

We do not need $J(l(p))$ since the function T is self-delimiting. However, the real analogon of Theorem 1.4.3 is the Coding Theorem, to be proved later in this section.

Part b of Theorem 1.3.2 holds for H, but is not the strongest what can be said. The counterpart of this theorem is the inequality (1.6.10) below. Theorem 1.4.1 and its corollary hold without change for H. The additivity relations will be proved in a sharper form for H in the next section.

1.6.2 Universal semimeasure

Self-delimiting complexity has an interesting characterization that is very useful in applications.

Definition 1.6.6 A function $w : S \to [0, 1]$ is called a *semimeasure* over the space S if

$$\sum_x w(x) \leq 1. \tag{1.6.5}$$

It is called a *measure* (a probability distribution) if equality holds here.

A semimeasure is called *constructive* if it is lower semicomputable. A constructive semimeasure which multiplicatively dominates every other constructive semimeasure is called *universal*. ⌟

Remark 1.6.7 Just as in an earlier remark, we warn that semimeasures will later be defined over the space $\mathbb{N}^{\mathbb{N}}$. They will be characterized by a nonnegative function w over $\mathbb{S} = \mathbb{N}^*$ but with a condition different from (1.6.5). ⌟

The following remark is useful.

Proposition 1.6.8 *If a constructive semimeasure w is also a measure then it is computable.*

Proof. If we compute an approximation w_t of the function w from below for which $1 - \varepsilon < \sum_x w_t(x)$ then we have $|w(x) - w_t(x)| < \varepsilon$ for all x. □

Theorem 1.6.3 *There is a universal constructive semimeasure.*

Proof. Every computable family w_i $(i = 1, 2, \dots)$ of semimeasures is dominated by the semimeasure $\sum_i 2^{-i} w_i$. Since there are only countably many constructive semimeasures there exists a semimeasure dominating them all. The only point to prove is that the dominating semimeasure can be made constructive.

Below, we construct a function $\mu_p(x)$ lower semicomputable in p, x such that μ_p as a function of x is a constructive semimeasure for all p and all constructive semimeasures occur among the w_p. Once we have μ_p we are done because for any positive computable function $\delta(p)$ with $\sum_p \delta(p) \leq 1$ the semimeasure $\sum_p \delta(p)\mu_p$ is obviously lower semicomputable and dominates all the μ_p.

Let $S_p(x)$ be the lower semicomputable function with Gödel number p and let $S_p^t(x)$ be a function recursive in p, t, x with rational values, dondecreasing in t, such that $\lim_t S_p^t(x) = \max\{0, S_p(x)\}$ and for each t, p, the function $S_p^t(x)$ is different from 0 only for $x \leq t$. Let us define μ_p^t recursively in t as follows. Let $\mu_p^0(x) = 0$. Suppose that μ_p^t is already defined. If S_p^{t+1} is a semimeasure then we define $\mu_p^{t+1} = S_p^{t+1}$, otherwise $\mu_p^{t+1} = \mu_p^t$. Let $\mu_p = \lim_t \mu_p^t$. The function $\mu_p(x)$ is, by its definition, lower semicomputable in p, x and a semimeasure for

each fixed p. It is equal to S_p whenever S_p is a semimeasure, hence it enumerates all constructive semimeasures. □

The above theorem justifies the following notation.

Definition 1.6.9 We choose a fixed universal constructive semimeasure and call it $\mathbf{m}(x)$. ⌟

With this notation, we can make it a little more precise in which sense this measure dominates all other constructive semimeasures.

Definition 1.6.10 For an arbitrary constructive semimeasure ν let us define

$$\mathbf{m}(\nu) = \sum \{\mathbf{m}(p) : \text{the (self-delimiting) program } p \text{ computes } \nu\}. \tag{1.6.6}$$

Similar notation will be used for other objects: for example now $\mathbf{m}(f)$ make sense for a recursive function f. ⌟

Let us apply the above concept to some interesting cases.

Theorem 1.6.4 *For all constructive semimeasures ν and for all strings x, we have*

$$\mathbf{m}(\nu)\nu(x) \stackrel{*}{<} \mathbf{m}(x). \tag{1.6.7}$$

Proof. Let us repeat the proof of Theorem 1.6.3, using $\mathbf{m}(p)$ in place of $\delta(p)$. We obtain a new constructive semimeasure $\mathbf{m}'(x)$ with the property $\mathbf{m}(\nu)\nu(x) \leq \mathbf{m}'(x)$ for all x. Noting $\mathbf{m}'(x) \stackrel{*}{=} \mathbf{m}(x)$ finishes the proof. □

Further, let us use the notation

$$H(x) = -\log \mathbf{m}(x)$$

for all kinds of argument x.

1.6.3 Prefix codes

The main theorem of this section says $H(x) \stackrel{+}{=} -\log \mathbf{m}(x)$. Before proving it, we relate descriptional complexity to the classical coding problem of information theory.

Definition 1.6.11 A (binary) *code* is any mapping from a set E of binary sequences to a set of objects. This mapping is the *decoding function* of the code. If the mapping is one-to-one, then sometimes the set E itself is also called a code.

A code is a *prefix code* if E is a prefix-free set. ⌟

The interpreter $U(p)$ is a decoding function.

Definition 1.6.12 We call string p the first shortest description of x if it is the lexicographically first binary word q with $|q| = K(x)$ and $U(q) = x$. ⌟

The set of first shortest descriptions is a code. This imposes the implicit lower bound stated in (1.5.1) on $K(x)$:

$$|\{x : K(x) = n\}| \leq 2^n. \tag{1.6.8}$$

The least shortest descriptions in the definition of $H(x)$ form moreover a *prefix code*.

We recall a classical result of information theory.

Lemma 1.6.13 (Kraft's Inequality, see [13]) *For any sequence $l_1, l_2, \ldots$ of natural numbers, there is a prefix code with exactly these numbers as codeword lengths, if and only if*

$$\sum_i 2^{-l_i} \leq 1. \tag{1.6.9}$$

Proof. Recall the standard correspondence $x \leftrightarrow [x]$ between binary strings and binary subintervals of the interval $[0, 1]$. A prefix code corresponds to a set of disjoint intervals, and the length of the interval $[x]$ is $2^{-l(x)}$. This proves that (1.6.9) holds for the lengths of codewords of a prefix code.

Suppose that l_i is given and (1.6.9) holds. We can assume that the sequence l_i is nondecreasing. Chop disjoint, adjacent intervals $I_1, I_2, \ldots$ of length $2^{-l_1}, 2^{-l_2}, \ldots$ from the left end of the interval $[0, 1]$. The right end of I_k is $\sum_{j=1}^{k} 2^{-l_j}$. Since the sequence l_j is nondecreasing, all intervals I_j are binary intervals. Take the binary string corresponding to I_j as the j-th codeword. □

Corollary 1.6.14 *We have* $-\log \mathbf{m}(x) \stackrel{+}{<} H(x)$.

Proof. The lemma implies

$$\sum_y 2^{-H(y|x)} \leq 1. \tag{1.6.10}$$

Since $H(x)$ is an upper semicomputable function, the function $2^{-H(x)}$ is lower semicomputable. Hence it is a constructive semimeasure, and we have $-\log \mathbf{m}(x) \stackrel{+}{<} H(x)$. □

The construction in the second part of the proof of Lemma 1.6.13 has some disadvantages. We do not always want to rearrange the numbers l_j, for example because we want that the order of the codewords reflect the order of our original objects. Without rearrangement, we can still achieve a code with only slightly longer codewords.

Lemma 1.6.15 (Shannon-Fano code) *(see [13]) Let $w_1, w_2, \ldots$ be positive numbers with $\sum_j w_j \leq 1$. There is a binary prefix code $p_1, p_2, \ldots$ where the codewords are in lexicographical order, such that*

$$|p_j| \leq -\log w_j + 2. \tag{1.6.11}$$

Proof. We follow the construction above and cut off disjoint, adjacent (not necessarily binary) intervals I_j of length w_j from the left end of $[0, 1]$. Let v_j be the length of the longest binary intervals contained in I_j. Let p_j be the binary word corresponding to the first one of these. Four or fewer intervals of length v_j cover I_j. Therefore (1.6.11) holds. □

1.6.4 The coding theorem for $H(x)$

The above results suggest $H(x) \leq -\log \mathbf{m}(x)$. But in the proof, we have to deal with the problem that $\mathbf{m}$ is not computable.

Theorem 1.6.5 (Coding Theorem, see [30, 17, 10])

$$H(x) \stackrel{+}{=} -\log \mathbf{m}(x) \tag{1.6.12}$$

Proof. We construct a self-delimiting p.r. function $F(p)$ with the property that $K_F(x) \leq -\log \mathbf{m}(x)+4$. The function F to be constructed is the decoding function of a prefix code hence the code-construction of Lemma 1.6.13 proposes itself. But since the function $\mathbf{m}(x)$ is only lower semicomputable, it is given only by a sequence converging to it from below.

Let $\{(z_t, k_t) : t = 1, 2, \ldots\}$ be a recursive enumeration of the set $\{(x, k) : k < \mathbf{m}(x)\}$ without repetition. Then

$$\sum_t 2^{-k_t} = \sum_x \sum_{z_t = 1} 2^{-k_t} \leq \sum_x 2\mathbf{m}(x) < 2.$$

Let us cut off consecutive adjacent, disjoint intervals I_t of length 2^{-k_t-1} from the left side of the interval $[0, 1]$. We define F as follows. If $[p]$ is a largest binary subinterval of some I_t then $F(p) = z_t$. Otherwise $F(p)$ is undefined.

The function F is obviously self-delimiting and partial recursive. It follows from the construction that for every x there is a t with $z_t = x$ and $0.5\mathbf{m}(x) < 2^{-k_t}$. Therefore, for every x there is a p such that $F(p) = x$ and $|p| \leq -\log \mathbf{m}(x)+4$. □

The Coding Theorem can be straightforwardly generalized as follows. Let $f(x, y)$ be a lower semicomputable nonnegative function. Then we have

$$H(y \mid x) \stackrel{+}{<} -\log f(x, y) \tag{1.6.13}$$

for all x with $\sum_y f(x, y) \leq 1$. The proof is the same.

1.6.5 Algorithmic probability

We can interpret the Coding Theorem as follows. It is known in classical information theory that for any probability distribution $w(x)$ ($x \in S$) a binary prefix code $f_w(x)$ can be constructed such that $l(f_w(x)) \leq -\log w(x) + 1$. We learned that for computable, moreover, even for contstructive distributions w there is a *universal code* with a self-delimiting partial-recursive decoding function T independent of w such that for the codeword length $H(x)$ we have

$$H(x) \leq -\log w(x) + c_w.$$

Here, only the additive constant c_w depends on the distribution w.

Let us imagine the self-delimiting Turing machine $\mathcal{T}$ computing our interpreter $T(p)$. Every time the machine asks for a new bit of the description, we could toss a coin to decide whether to give 0 or 1.

Definition 1.6.16 Let $P_T(x)$ be the probability that the self-delimiting machine $\mathcal{T}$ gives out result x after receiving random bits as inputs. ⌟

We can write $P_T(x) = \sum W_T(x)$ where $W_T(x)$ is the set $\{2^{-l(p)} : T(p) = x\}$. Since $2^{-H(x)} = \max W_T(x)$, we have $-\log P_T(x) \leq H(x)$. The semimeasure P_T is constructive, hence $P \overset{*}{<} \mathbf{m}$, hence

$$H \overset{+}{<} -\log \mathbf{m} \overset{+}{<} -\log P_T \leq H,$$

hence

$$-\log P_T \overset{+}{=} -\log \mathbf{m} \overset{+}{=} H.$$

Hence the sum $P_T(x)$ of the set $W_T(x)$ is at most a constant times larger than its maximal element $2^{-H(x)}$. This relation was not obvious in advance. The outcome x might have high probability because it has many long descriptions. But we found that then it must have a short description too. In what follows it will be convenient to fix the definition of $\mathbf{m}(x)$ as follows.

Definition 1.6.17 From now on let us define

$$\mathbf{m}(x) = P_T(x). \tag{1.6.14}$$

⌟

1.7 The statistics of description length

We can informally summarize the relation $H \overset{+}{=} -\log P_T$ saying that *if an object has many long descriptions then it has a short one*. But how many descriptions does an

object really have? With the Coding Theorem, we can answer several questions of this kind. The reader can view this section as a series of exercises in the technique acquired up to now.

Theorem 1.7.1 *Let $f(x, n)$ be the number of binary strings p of length n with $T(p) = x$ for the universal s.d. interpreter T defined in Theorem 1.6.1. Then for every $n \geq H(x)$, we have*

$$\log f(x, n) \stackrel{+}{=} n - H(x, n). \tag{1.7.1}$$

Using $H(x) \stackrel{+}{<} H(x, n)$, which holds just as (1.4.1), and substituting $n = H(x)$ we obtain $\log f(x, H(x)) \stackrel{+}{=} H(x) - H(x, H(x)) \stackrel{+}{<} 0$, implying the following.

Corollary 1.7.1 *The number of shortest descriptions of any object is bounded by a universal constant.*

Since the number $\log f(x, H(x))$ is nonnegative, we also derived the identity

$$H(x, H(x)) \stackrel{+}{=} H(x) \tag{1.7.2}$$

which could have been proven also in the same way as Theorem 1.4.4.

Proof of Theorem 1.7.1. It is more convenient to prove equation (1.7.1) in the form

$$2^{-n} f(x, n) \stackrel{*}{=} \mathbf{m}(x, n) \tag{1.7.3}$$

where $\mathbf{m}(x, y) = \mathbf{m}(\langle x, y \rangle)$. First we prove $\stackrel{*}{<}$. Let us define a p.r. self-delimiting function F by $F(p) = \langle T(p), l(p) \rangle$. Applying F to a coin-tossing argument, $2^{-n} f(x, n)$ is the probability that the pair $\langle x, n \rangle$ is obtained. Therefore the left-hand side of (1.7.3), as a function of $\langle x, n \rangle$, is a constructive semimeasure, dominated by $\mathbf{m}(x, n)$.

Now we prove $\stackrel{*}{>}$. We define a self-delimiting p.r. function G as follows. The machine computing $G(p)$ tries to decompose p into three segments $p = \beta(c)^o v w$ in such a way that $T(v)$ is a pair $\langle x, l(p) + c \rangle$. If it succeeds then it outputs x. By the universality of T, there is a binary string g such that $T(gp) = G(p)$ for all p. Let $r = l(g)$. For an arbitrary pair x, n, let q be a shortest binary string with $T(q) = \langle x, n \rangle$, and w an arbitrary string of length

$$l = n - l(q) - r - l(\beta(r)^o).$$

Then $G(\beta(r)^o q w) = T(g \beta(r)^o q w) = x$. Since w is arbitrary here, there are 2^l binary strings p of length n with $T(p) = x$. □

How many objects are there with a given complexity n? We can answer this question with a good approximation.

Definition 1.7.2 Let $g_T(n)$ be the number of objects $x \in \mathbb{S}$ with $H(x) = n$, and D_n the set of binary strings p of length n for which $T(p)$ is defined. Let us define the moving average

$$h_T(n, c) = \frac{1}{2c+1} \sum_{i=-c}^{c} g_T(n+i).$$

⌟

Here is an estimation of these numbers.

Theorem 1.7.2 *([46]) There is a natural number c such that*

$$\log |D_n| \stackrel{+}{=} n - H(n), \tag{1.7.4}$$

$$\log h_T(n, c) \stackrel{+}{=} n - H(n). \tag{1.7.5}$$

Since we are interested in general only in accuracy up to additive constants, we could omit the normalizing factor $1/(2c+1)$ from the moving average h_T. We do not know whether this average can be replaced by $g_T(n)$. This might depend on the special universal partial recursive function we use to construct the optimal s.d. interpreter $T(p)$. But equation (1.7.5) is true for any *optimal* s.d. interpreter T (such that the inequality (1.6.1) holds for all F) while there is an optimal s.d. interpreter F for which $g_F(n) = 0$ for every odd n. Indeed, let $F(00p) = T(p)$ if $l(p)$ is even, $F(1p) = T(p)$ if $l(p)$ is odd and let F be undefined in all other cases. Then F is defined only for inputs of even length, while $K_F \le K_T + 2$.

Lemma 1.7.3

$$\sum_y \mathbf{m}(x, y) \stackrel{*}{=} \mathbf{m}(x).$$

Proof. The left-hand side is a constructive semimeasure therefore it is dominated by the right side. To show $\stackrel{*}{>}$, note that by equation (1.4.2) as applied to H, we have $H(x, x) \stackrel{+}{=} H(x)$. Therefore $\mathbf{m}(x) \stackrel{*}{=} \mathbf{m}(x, x) < \sum_y \mathbf{m}(x, y)$. □

Proof of Theorem 1.7.2: Let $d_n = |D_n|$. Using Lemma 1.7.3 and Theorem 1.7.1 we have

$$g_T(n) \le d_n = \sum_x f(x, n) \stackrel{*}{=} 2^n \sum_x \mathbf{m}(x, n) \stackrel{*}{=} 2^n \mathbf{m}(n).$$

Since the complexity H has the same continuity property (1.4.3) as K, we can derive from here $h_T(n, c) \le 2^n \mathbf{m}(n) O(2^c c^2)$. To prove $h_T(n, c) \stackrel{*}{>} d_n$ for an appropriate c, we will prove that the complexity of at least half of the elements of D_n is near n.

For some constant c_0, we have $H(p) \le n + c_0$ for all $p \in D_n$. Indeed, if the s.d. p.r. interpreter $F(p)$ is equal to p where $T(p)$ is defined and is undefined

otherwise then $K_F(p) \leq n$ for $p \in D_n$. By $H(p, l(p)) \stackrel{+}{=} H(p)$, and a variant of Lemma 1.7.3, we have

$$\sum_{p \in D_n} \mathbf{m}(p) \stackrel{*}{=} \sum_{p \in D_n} \mathbf{m}(p, n) \stackrel{*}{=} \mathbf{m}(n).$$

Using the expression (1.7.4) for d_n, we see that there is a constant c_1 such that

$$d_n^{-1} \sum_{p \in D_n} 2^{-H(p)} \leq 2^{-n+c_1}.$$

Using Markov's Inequality, hence the number of elements p of D_n with $H(p) \geq n - c_1 - 1$ is at least $d_n/2$. We finish the proof making c to be the maximium of c_0 and $c_1 + 1$. □

What can be said about the complexity of a binary string of length n? We mentioned earlier the estimate $H(x) \stackrel{+}{<} n + 2 \log n$. The same argument gives the stronger estimate

$$H(x) \stackrel{+}{<} l(x) + H(l(x)). \tag{1.7.6}$$

By a calculation similar to the proof of Theorem 1.7.2, we can show that the estimate (1.7.6) is exact for most binary strings. First, it follows just as in Lemma 1.7.3 that there is a constant c such that

$$2^{-n} \sum_{l(p)=n} 2^{-H(p)} \leq c 2^{-n} \mathbf{m}(n).$$

From this, Markov's Inequality gives that the number of strings $p \in \mathbb{B}^n$ with $H(p) < n - k$ is at most $c 2^{n-k}$.

There is a more polished way to express this result.

Definition 1.7.4 Let us introduce the following function of natural numbers:

$$H^+(n) = \max_{k \leq n} H(k).$$

⌟

This is the smallest monotonic function above $H(n)$. It is upper semicomputable just as H is, hence 2^{-H^+} is universal among the monotonically decreasing constructive semimeasures. The following theorem expresses H^+ more directly in terms of H.

Theorem 1.7.3 *We have* $H^+(n) \stackrel{+}{=} \log n + H(\lfloor \log n \rfloor)$.

Proof. To prove $\overset{+}{<}$, we construct a s.d. interpreter as follows. The machine computing $F(p)$ finds a decomposition uv of p such that $T(u) = l(v)$, then outputs the number whose binary representation (with possible leading zeros) is the binary string v. With this F, we have

$$H(k) \overset{+}{<} K_F(k) \leq \log n + H(\lfloor \log n \rfloor) \tag{1.7.7}$$

for all $k \leq n$. The numbers between $n/2$ and n are the ones whose binary representation has length $\lfloor \log n \rfloor + 1$. Therefore if the bound (1.7.6) is sharp for most x then the bound (1.7.7) is sharp for most k. □

The sharp monotonic estimate $\log n + H(\lfloor \log n \rfloor)$ for $H(n)$ is less satisfactory than the sharp estimate $\log n$ for Kolmogorov's complexity $K(n)$, because it is not a computable function. We can derive several computable upper bounds from it, for example $\log n + 2 \log\log n$, $\log n + \log\log n + 2 \log\log\log n$, etc. but none of these is sharp for large n. We can still hope to find computable upper bounds of H which are sharp infinitely often. The next theorem provides one.

Theorem 1.7.4 (see [46]) *There is a computable upper bound $G(n)$ of the function $H(n)$ with the property that $G(n) = H(n)$ holds for infinitely many n.*

For the proof, we make use of a s.d. Turing machine $\mathcal{T}$ computing the function T. Similarly to the proof of Theorem 1.5.1, we introduce a time-restricted complexity.

Definition 1.7.5 We know that for some constant c, the function $H(n)$ is bounded by $2 \log n + c$. Let us fix such a c.

For any number n and s.d. Turing machine $\mathcal{M}$, let $H_{\mathcal{M}}(n; t)$ be the minimum of $2 \log n + c$ and the lengths of descriptions from which $\mathcal{M}$ computes n in t or fewer steps. ⌟

At the time we proved the Invariance Theorem we were not interested in exactly how the universal p.r. function V was to be computed. Now we need to be more specific.

Definition 1.7.6 Let us agree now that the universal p.r. function is computed by a universal Turing machine $\mathcal{V}$ with the property that for any Turing machine $\mathcal{M}$ there is a constant m such that the number of steps needed to simulate t steps of $\mathcal{M}$ takes no more than mt steps on $\mathcal{V}$. Let $\mathcal{T}$ be the Turing machine constructed from $\mathcal{V}$ computing the optimal s.d. interpreter T, and let us write

$$H(n, t) = H_{\mathcal{T}}(n, t).$$

⌟

With these definitions, for each s.d. Turing machine $\mathcal{M}$ there are constants m_0, m_1 with

$$H(n, m_0 t) \le H_{\mathcal{M}}(n, t) + m_1. \tag{1.7.8}$$

The function $H(n; t)$ is nonincreasing in t.

Proof of Theorem 1.7.4. First we prove that there are constants c_0, c_1 such that $H(n; t) < H(n; t-1)$ implies

$$H(\langle n, t\rangle; c_0 t) \le H(n; t) + c_1. \tag{1.7.9}$$

Indeed, we can construct a Turing machine $\mathcal{M}$ simulating the work of $\mathcal{T}$ such that if $\mathcal{T}$ outputs a number n in t steps then $\mathcal{M}$ outputs the pair $\langle n, t\rangle$ in $2t$ steps:

$$H_{\mathcal{M}}(\langle n, t\rangle; 2t) \le H(n, t).$$

Combining this with the inequality (1.7.8) gives the inequality (1.7.9). We define now a recursive function F as follows. To compute $F(x)$, the s.d. Turing machine first tries to find n, t such that $x = \langle n, t\rangle$. (It stops even if it did not find them.) Then it outputs $H(x; c_0 t)$. Since $F(x)$ is the length of some description of x, it is an upper bound for $H(x)$. On the other hand, suppose that x is one of the infinitely many integers of the form $\langle n, t\rangle$ with

$$H(n) = H(n; t) < H(n; t-1).$$

Then the inequality (1.7.9) implies $F(x) \le H(n) + c_1$ while $H(n) \stackrel{+}{<} H(x)$ is known (it is the equivalent of 1.4.4 for H), so $F(x) \le H(n) + c$ for some constant c. Now if F would not be equal to H at infinitely many places, only close to it (closer than c) then we can decrease it by some additive constant less than c and modify it at finitely many places to obtain the desired function G. □

2 Randomness

2.1 Uniform distribution

Speaking of a "random binary string", one generally understands randomness with respect to the coin-toss distribution (when the probability $P(x)$ of a binary sequence x of length n is 2^{-n}). This is the only distribution considered in the present section.

One can hope to distinguish sharply between random and nonrandom *infinite sequences*. Indeed, an infinite binary sequence whose elements are 0 with only finitely many exceptions, can be considered nonrandom without qualification. This section defines randomness for finite strings. For a sequence x of length n, it would be unnatural to fix some number k, declare x nonrandom when its first k elements are 0, but permit it to be random if only the first $k-1$ elements are 0. So, we will just declare some finite strings less random than others. For this, we introduce a certain real-valued function $d(x) \geq 0$ measuring the *deficiency of randomness* in the string x. The occurrence of a nonrandom sequence is considered an exceptional event, therefore the function $d(x)$ can assume large values only with small probability. What is "large" and "small" depends only on our "unit of measurement". We require for all n, k

$$\sum \{P(x) : x \in \mathbb{B}^n,\ d(x) > k\} < 2^{-k}, \tag{2.1.1}$$

saying that there be at most 2^{n-k} binary sequences x of length n with $d(x) > k$. Under this condition, we even allow $d(x)$ to take the value ∞.

To avoid arbitrariness in the distinction between random and nonrandom, the function $d(x)$ must be *simple*. We assume therefore that the set $\{(n, k, x) : l(x) = n,\ d(x) > k\}$ is recursively enumerable, or, which is the same, that the function

$$\mathbf{d} : \Omega \to (-\infty, \infty]$$

is lower semicomputable.

Remark 2.1.1 We do not assume that the set of strings $x \in \mathbb{B}^n$ with *small* deficiency of randomness is enumerable, because then it would be easy to construct a "random" sequence. ⌟

Definition 2.1.2 A lower semicomputable function $d : \mathbb{S}_2 \to \mathbb{R}$ (where $\mathbb{R}$ is the set of real numbers) is called a *Martin-Löf test (ML-test)*, or a *probability-bounded test* if it satisfies (2.1.1).

A ML-test $d_0(x)$ is *universal* if it additively dominates all other ML-tests: for any other ML-test $d(x)$ there is a constant $c < \infty$ such that for all x we have $d(x) < d_0(x) + c$. ⌟

If a test d_0 is universal then any other test d of randomness can discover at most by a constant amount more deficiency of randomness in any sequence x than $d_0(x)$. Obviously, the difference of any two universal ML-tests is bounded by some constant.

The following theorem reveals a simple connnection between descriptional complexity and a certain randomness property.

Definition 2.1.3 Let us define the following function for binary strings x:

$$d_0(x) = l(x) - K(x \mid l(x)). \tag{2.1.2}$$

⌟

Theorem 2.1.1 (Martin-Löf) *The function $d_0(x)$ is a universal Martin-Löf test.*

Proof. Since $K(x \mid y)$ is semicomputable from above, d_0 is semicomputable from below. The property of semicomputability holds for d_0 as a straightforward consequence of Theorem 1.5.1 Therefore d_0 is a ML-test. We must show that it is larger (to within an additive constant) than any other ML-test. Let d be a ML-test. Let us define the function $F(x, y)$ to be $y - d(x)$ for $y = l(x)$, and ∞ otherwise. Then F is upper semicomputable, and satisfies (1.5.1). Therefore by Theorem 1.5.3, we have $K(x \mid l(x)) \stackrel{+}{<} l(x) - d(x)$. □

Theorem 2.1.1 says that under very general assumptions about randomness, those strings x are random whose descriptional complexity is close to its maximum, $l(x)$. The more "regularities" are discoverable in a sequence, the less random it is. However, this is true only of regularities which decrease the descriptional complexity. "Laws of randomness" are regularities whose probability is high, for example the law of large numbers, the law of iterated logarithm, the arcsine law, etc. A random sequence will satisfy *all such laws*.

Example 2.1.4 The law of large numbers says that in most binary sequences of length n, the number of 0's is close to the numer of 1's. We prove this law of

probability theory by constructing a ML-test $d(x)$ taking large values on sequences in which the number of 0's is far from the number of 1's.

Instead of requiring (2.1.1) we require a somewhat stronger property of $d(x)$: for all n,

$$\sum_{x\in\mathbb{B}^n} P(x)2^{d(x)} \leq 1. \tag{2.1.3}$$

From this, the inequality (2.1.1) follows by the following well-known inequality called the Markov Inequality which says the following. Let P be any probability distribution, f any nonnegative function, with expected value $E_P(f) = \sum_x P(x)f(x)$. For all $\lambda \geq 0$ we have

$$\sum\{P(x) : f(x) > \lambda E_P(f)\} \leq \lambda. \tag{2.1.4}$$

For any string $x \in \mathbb{S}$, and natural number i, let $N(i \mid x)$ denote the number of occurrences of i in x. For a binary string x of length n define $p_x = N(1 \mid x)/n$, and

$$P_x(y) = p_x^{N(1|y)}(1-p_x)^{N(0|y)}$$
$$d(x) = \log P_x(x) + n - \log(n+1).$$

We show that $d(x)$ is a ML-test. It is obviously computable. We prove (2.1.3). We have

$$\sum_x P(x)2^{d(x)} = \sum_x 2^{-n}2^n P_x(x)\frac{1}{n+1} = \frac{1}{n+1}\sum_x P_x(x)$$
$$= \frac{1}{n+1}\sum_{k=0}^{n}\binom{n}{k}(\frac{k}{n})^k(1-\frac{k}{n})^{n-k} < \frac{1}{n+1}\sum_{k=0}^{n} 1 = 1.$$

The test $d(x)$ expresses the (weak) law of large numbers in a rather strong version. We rewrite $d(x)$ as

$$d(x) = n(1-h(p_x)) - \log(n+1)$$

where $h(p) = -p\log p - (1-p)\log(1-p)$. The entropy function $h(p)$ achieves its maximum 1 at $p = 1/2$. Therefore the test $d(x)$ tells us that the probability of sequences x with $p_x < p < 1/2$ for some constant p is bounded by the exponentially decreasing quantitiy $(n+1)2^{-n(1-h(p))}$. We also see that since for some constant c we have

$$1 - h(p) > c(p-1/2)^2,$$

therefore if the difference $|p_x - 1/2|$ is much larger than

$$\sqrt{\frac{\log n}{cn}}$$

then the sequence x is "not random", and hence, by Theorem 2.1.1 its complexity is significantly smaller than n. ⌟

2.2 Computable distributions

Let us generalize the notion of randomness to arbitrary discrete computable probability distributions.

2.2.1 Two kinds of test

Let $P(x)$ be a probability distribution over some discrete countable space Ω, which we will identify for simplicity with the set $\mathbb{S}$ of finite strings of natural numbers. Thus, P is a nonnegative function with

$$\sum_x P(x) = 1.$$

Later we will consider probability distributions P over the space $\mathbb{N}^{\mathbb{N}}$, and they will also be characterized by a nonnegative function $P(x)$ over $\mathbb{S}$. However, the above condition will be replaced by a different one.

We assume P to be computable, with some Gödel number e. We want to define a test $d(x)$ of randomness with respect to P. It will measure how justified is the assumption that x is the outcome of an experiment with distribution P.

Definition 2.2.1 A function $d : \mathbb{S} \to \mathbb{R}$ is an *integrable test*, or *expectation-bounded test* of randomness with respect to P if it is lower semicomputable and satisfies the condition

$$\sum_x P(x)2^{d(x)} \leq 1. \tag{2.2.1}$$

It is *universal* if it dominates all other integrable tests to within an additive constant. ⌟

(A similar terminology is used in [35] in order to distinguish tests satisfying condition (2.2.1) from Martin-Löf tests.)

Remark 2.2.2 We must allow $d(x)$ to take negative values, even if only, say values ≥ -1, since otherwise condition (2.2.1) could only be satisfied with $d(x) = 0$ for all x with $P(x) > 0$. ⌟

Proposition 2.2.3 *Let $c = \log(2\pi^2/6)$. If a function $d(x)$ satisfies (2.2.1) then it satisfies (2.1.1). If $d(x)$ satisfies (2.1.1) then $d - 2\log d - c$ satisfies (2.2.1).*

Proof. The first statement follows by Markov's Inequality (2.1.4). The second one can be checked by immediate computation. □

Thus, condition 2.2.1 is nearly equivalent to condition (2.1.1): to each test d according to one of them, there is a test d' asymptotically equal to d satisfying the other. But condition (2.2.1) has the advantage of being just one inequality instead of the infinitely many ones.

2.2.2 Randomness via complexity

We want to find a universal integrable test. Let us introduce the function $w(x) = P(x)2^{d(x)}$. The above conditions on $d(x)$ imply that $w(x)$ is semicomputable from below and satisfies $\sum_x w(x) \le 1$: so, it is a constructive semimeasure. With the universal constructive semimeasure of Theorem 1.6.3, we also obtain a universal test for any computable probability distribution P.

Definition 2.2.4 For an arbitrary measure P over a discrete space Ω, let us denote

$$\overline{\mathbf{d}}_P(x) = \log \frac{\mathbf{m}(x)}{P(x)} = -\log P(x) - H(x). \tag{2.2.2}$$

⌟

The following theorem shows that randomness can be tested by checking how close is the complexity $H(x)$ to its upper bound $-\log P(x)$.

Theorem 2.2.1 *The function $\overline{\mathbf{d}}_P(x) = -\log P(x) - H(x)$ is a universal integrable test for any fixed computable probability distribution P. More exactly, it is lower semicomputable, satisfies* (2.2.1) *and for all integrable tests $d(x)$ for P, we have*

$$d(x) \stackrel{+}{<} \overline{\mathbf{d}}_P(x) + H(d) + H(P).$$

Proof. Let $d(x)$ be an integrable test for P. Then $\nu(x) = 2^{d(x)}P(x)$ is a constructive semimeasure, and it has a self-delimiting program of length $\stackrel{+}{<} H(d)+H(P)$. It follows (using the definition (1.6.6)) that $m(\nu) \stackrel{*}{<} 2^{H(d)+H(P)}$; hence inequality (1.6.7) gives

$$\nu(x)2^{H(d)+H(P)} \stackrel{*}{<} \mathbf{m}(x).$$

Taking logarithms finishes the proof. □

The simple form of our universal test suggests remarkable interpretations. It says that the outcome x is random with respect to the distribution P, if the latter assigns to x a large enough probability—however, not in absolute terms, only relatively to the universal semimeasure $\mathbf{m}(x)$. The relativization is essential, since otherwise we could not distinguish between random and nonrandom outcomes for the uniform distribution P_n defined in Section 2.1.

For any (not necessarily computable) distribution P, the P-expected value of the function $\mathbf{m}(x)/P(x)$ is at most 1, therefore the relation

$$\mathbf{m}(x) \le kP(x)$$

holds with probability not smaller than $1 - 1/k$. If P is computable then the inequality

$$\mathbf{m}(P)P(x) \stackrel{*}{<} \mathbf{m}(x)$$

holds for all x. Therefore the following applications are obtained.

- If we assume x to be the outcome of an experiment with some simple computable probability distribution P then $\mathbf{m}(x)$ is a good estimate of $P(x)$. The goodness depends on how simple P is to define and how random x is with respect to P: how justified is the assumption. Of course, we cannot compute $\mathbf{m}(x)$ but it is nevertheless well defined.
- If we trust that a given object x is random with respect to a given distribution P then we can use $P(x)$ as an estimate of $\mathbf{m}(x)$. The degree of approximation depends on the same two factors.

Let us illustrate the behavior of the universal semimeasure $\mathbf{m}(x)$ when x runs over the set of natural numbers. We define the semimeasures v and w as follows. Let $v(n) = c/n^2$ for an appropriate constant c, let $w(n) = v(x)$ if $n = 2^{2^k}$, and 0 otherwise. Then $\mathbf{m}(n)$ dominates both $v(n)$ and $w(n)$. The function $\mathbf{m}(n)$ dominates $1/n \log^2 n$, but jumps at many places higher than that. Indeed, it is easy to see that $\mathbf{m}(n)$ converges to 0 slower than any positive computable function converging to 0: in particular, it is not computable. Hence it cannot be a measure, that is $\sum_x \mathbf{m}(x) < 1$. We feel that we can make $\mathbf{m}(x)$ "large" whenever x is "simple".

We cannot compare the universal test defined in the present section directly with the one defined in Section 2.1. There, we investigated tests for a whole family P_n of distributions, where P_n is the uniform distribution on the set $\mathbb{B}^n$. That the domain of these distributions is a different one for each n is not essential since we can identify the set $\mathbb{B}^n$ with the set $\{2^n, \ldots, 2^{n+1} - 1\}$. Let us rewrite the function $d_0(x)$ defined in (2.1.2) as

$$d_0(x) = n - K(x \mid n)$$

for $x \in \mathbb{B}^n$, and set it ∞ for x outside $\mathbb{B}^n$. Now the similarity to the expression

$$\overline{\mathbf{d}}_P(x) = -\log P(x) - \log \mathbf{m}(x)$$

is striking because $n = \log P_n(x)$. Together with the remark made in the previous paragraph, this observation suggests that the value of $-\log \mathbf{m}(x)$ is close to the complexity $K(x)$.

2.2.3 Conservation of randomness

The idea of conservation of randomness has been expressed first by Levin, who has published some papers culminating in [32] developing a general theory. In this section, we address the issue in its simplest form only. Suppose that we want to talk about the randomness of integers. One and the same integer x can be represented in decimal, binary, or in some other notation: this way, every time a different string will represent the same number. Still, the form of the representation should not affect the question of randomness of x significantly. How to express this formally?

Let P be a computable measure and let $f : \mathbb{S} \to \mathbb{S}$ be a computable function. We expect that under certain conditions, the randomness of x should imply the randomness of $f(x)$—but, for which distribution? For example, if x is a binary string and $f(x)$ is the decimal notation for the number expressed by $1x$ then we expect $f(x)$ to be random not with respect to P, but with respect to the measure f^*P that is the image of P under f. This measure is defined by the formula

$$(f^*P)(y) = P(f^{-1}(y)) = \sum_{f(x)=y} P(x). \tag{2.2.3}$$

Proposition 2.2.5 *If f is a computable function and P is a computable measure then f^*P is a computable measure.*

Proof. It is sufficient to show that f^*P is lower semicomputable, since we have seen that a lower semicomputable measure is computable. But, the semicomputability can be seen immediately from the definition. □

Let us see that randomness is indeed preserved for the image. Similarly to (1.6.6), let us define for an arbitrary computable function f:

$$\mathbf{m}(f) = \sum \{\mathbf{m}(p) : \text{program } p \text{ computes } f\}. \tag{2.2.4}$$

Theorem 2.2.2 *Let f be a computable function. Between the randomness test for P and the test for the image of P, the following relation holds, for all x:*

$$\overline{\mathbf{d}}_{f^*P}(f(x)) \stackrel{+}{<} \overline{\mathbf{d}}_P(x) + H(f) + H(P). \tag{2.2.5}$$

Proof. Let us denote the function on the left-hand side of (2.2.5) by

$$d_P(x) = \overline{\mathbf{d}}_{f^*P}(f(x)) = \log \frac{\mathbf{m}(f(x))}{(f^*P)(f(x))}.$$

It is lower semicomputable, with the help of a program of length $H(f) + H(P)$, by its very definition. Let us check that it is an integrable test, so it satisfies the inequality (2.2.1). We have

$$\sum_x P(x)2^{d_P(x)} = \sum_y (f^*P)(y)2^{\overline{\mathbf{d}}_{f^*P}(y)} = \sum_y \mathbf{m}(y) \le 1.$$

Hence $d_P(x)$ is a test, and hence it is $\stackrel{+}{<} \overline{\mathbf{d}}_P(x) + H(f) + H(P)$, by the universality of the test $\overline{\mathbf{d}}_P(x)$. (Strictly speaking, we must modify the proof of Theorem 2.2.1 and see that a program of length $H(f) + H(P)$ is also sufficient to define the semimeasure $P(x)d(x \mid P)$.) □

The appearance of the term $H(f)$ on the right-hand side is understandable: using some very complex function f, we can certainly turn any string into any other one, also a random string into a much less random one. On the other hand, the term $H(P)$ appears only due to the imperfection of our concepts. It will disappear in the more advanced theory presented in later sections, where we develop *uniform tests*.

Example 2.2.6 For each string x of length $n \ge 1$, let

$$P(x) = \frac{2^{-n}}{n(n+1)}.$$

This distribution is uniform on strings of equal length. Let $f(x)$ be the function that erases the first k bits. Then for $n \ge 1$ and for any string x of length n we have

$$(f^*P)(x) = \frac{2^{-n}}{(n+k)(n+k+1)}.$$

(The empty string has the rest of the weight of f^*P.) The new distribution is still uniform on strings of equal length, though the total probability of strings of length n has changed somewhat. We certainly expect randomnes conservation here: after erasing the first k bits of a random string of length $n+k$, we should still get a random string of length n. ⌟

The computable function f applied to a string x can be viewed as some kind of transformation $x \mapsto f(x)$. As we have seen, it does not make x less random. Suppose now that we introduce randomization into the transformation process itself: for example, the machine computing $f(x)$ can also toss some coins. We want to say that randomness is also conserved under such more general transformations, but first of all, how to express such a transformation mathematically? We will describe it by a "matrix": a computable probability transition function $T(x, y) \ge 0$

with the property that

$$\sum_y T(x,y) = 1.$$

Now, the image of the distribution P under this transformation can be written as T^*P, and defined as follows:

$$(T^*P)(y) = \sum_x P(x)T(x,y).$$

How to express now randomness conservation? It is certainly not true that every possible outcome y is as random with respect to T^*P as x is with respect to P. We can only expect that for each x, the conditional probability (in terms of the transition $T(x,y)$) of those y whose non-randomness is larger than that of x, is small. This will indeed be expressed by the corollary below. To get there, we upperbound the $T(x,\cdot)$-expected value of $2^{\overline{\mathbf{d}}_{T^*P}(y)}$. Let us go over to exponential notation:

Definition 2.2.7 Denote

$$\overline{\mathbf{t}}_P(x) = 2^{\overline{\mathbf{d}}_P(x)} = \frac{\mathbf{m}(x)}{P(x)}. \tag{2.2.6}$$

⌟

Theorem 2.2.3 *We have*

$$\log \sum_y T(x,y)2^{\overline{\mathbf{d}}_{T^*P}(y)} \stackrel{+}{<} \overline{\mathbf{d}}_P(x) + H(T) + H(P). \tag{2.2.7}$$

Proof. Let us denote the function on the left-hand side of (2.2.7) by $t_P(x)$. It is lower semicomputable by its very construction, using a program of length $\stackrel{+}{<} H(T)+H(P)$. Let us check that it satisfies $\sum_x P(x)t_P(x) \le 1$ which, in this notation, corresponds to inequality (2.2.1). We have

$$\begin{aligned}\sum_x P(x)t_P(x) &= \sum_x P(x) \sum_y T(x,y)t_{T^*P}(y) \\ &= \sum_x P(x) \sum_y T(x,y)\frac{\mathbf{m}(y)}{(T^*P)(y)} = \sum_y \mathbf{m}(y) \le 1.\end{aligned}$$

It follows that $d_P(x) = \log t_P(x)$ is an integrable test and hence $d_P(x) \stackrel{+}{<} \overline{\mathbf{d}}_P(x) + H(T) + H(P)$. (See the remark at the end of the proof of Theorem 2.2.2.) □

Corollary 2.2.8 *There is a constant c_0 such that for every integer $k \ge 0$, for all x we have*

$$\sum\{T(x,y) : \overline{\mathbf{d}}_{T^*P}(y) - \overline{\mathbf{d}}_P(x) > k + H(T) + H(P)\} \le 2^{-k+c_0}.$$

Proof. The theorem says

$$\sum_y T(x,y)\bar{\mathbf{t}}_{T^*P}(y) \stackrel{*}{<} t_P(x)2^{H(T)+H(P)}.$$

Thus, it upperbounds the expected value of the function $2^{\bar{\mathbf{d}}_{T^*P)}(y)}$ according to the distribution $T(x,\cdot)$. Applying Markov's inequality (2.1.4) to this function yields the desired result. □

2.3 Infinite sequences

These lecture notes treat the theory of randomness over continuous spaces mostly separately, starting in Section 4.1. But the case of computable measures over infinite sequences is particularly simple and appealing, so we give some of its results here, even if most follow from the more general results given later.

In this section, we will fix a finite or countable *alphabet* $\Sigma = \{s_1, s_2, \ldots\}$, and consider probability distributions over the set

$$X = \Sigma^{\mathbb{N}}$$

of infinite sequences with members in Σ. An alternative way of speaking of this is to consider a sequence of *random variables* $X_1, X_2, \ldots$, where $X_i \in \Sigma$, with a joint distribution. The two interesting extreme special cases are $\Sigma = \mathbb{B} = \{0,1\}$, giving the set of infinite 0-1 sequences, and $\Sigma = \mathbb{N}$, giving the set sequences of natural numbers.

2.3.1 Null sets

Our goal is here to illustrate the measure theory developed in Section A.2, through Subsection A.2.3, on the concrete example of the set of infinite sequences, and then to develop the theory of randomness in it.

We will distinguish a few simple kinds of subsets of the set S of sequences, those for which probability can be defined especially easily.

Definition 2.3.1

- For string $z \in \Sigma^*$, we will denote by zX the set of elements of X with prefix z. Such sets will be called *cylinder sets*.
- A subset of X is *open* if it is the union of any number (not necessarily finite) of cylinder sets. It is called *closed* if its complement is open.
- An open set is called *constructive* if it is the union of a recursively enumerable set of cylinder sets.

- A set is called G_δ if it is the intersection of a sequence of open sets. It is called F_σ if it is the union of a sequence of closed sets.
- We say that a set $E \subseteq X$ is *finitely determined* if there is an n and an $\mathcal{E} \subseteq \Sigma^n$ such that

$$E = \mathcal{E}X = \bigcup_{s \in \mathcal{E}} sX.$$

  Let $\mathcal{F}$ be the class of all finitely determined subsets of X.
- A class of subsets of X is called an *algebra* if it is closed with respect to finite intersections and complements (and then of course, also with respect to finite unions). It is called a σ-algebra (sigma-algebra) when it is also closed with respect to countable intersections (and then of course, also with respect to countable unions).

⌟

Example 2.3.2 An example open set that is not finitely determined, is the set E of all sequences that contain a substring 11. ⌟

The following observations are easy to prove.

Proposition 2.3.3

a) *Every open set G can be represented as the union of a sequence of* disjoint *cylinder sets.*

b) *The set of open sets is closed with respect to finite intersection and arbitrarily large union.*

c) *Each finitely determined set is both open and closed.*

d) *The class $\mathcal{F}$ of finitely determined sets forms an algebra.*

Probability is generally defined as a nonnegative, monotonic function (a "measure", see later) on some subsets of the event space (in our case the set X) that has a certain additivity property. We do not need immediately the complete definition of measures, but let us say what we mean by an additive set function.

Definition 2.3.4 Consider a nonnegative, function μ is defined over some subsets of X. that is also *monotonic*, that is $A \subseteq B$ implies $\mu(A) \subseteq \mu(B)$. We say that it is *additive* if, whenever it is defined on the sets $E_1, \ldots, E_n$, and on $E = \bigcup_{i=1}^n E_i$ and the sets E_i are mutually disjoint, then we have $\mu(E) = \mu(E_1) + \cdots + \mu(E_n)$.

We say that μ is *countably additive*, or σ-additive (sigma-additive), if in addition whenever it is defined on the sets $E_1, E_2, \ldots$, and on $E = \bigcup_{i=1}^\infty E_i$ and the sets E_i are mutually disjoint, then we have $\mu(E) = \sum_{i=1}^\infty \mu(E_i)$. ⌟

First we consider only measures defined on cylinder sets.

Definition 2.3.5 (Measure over $\Sigma^{\mathbb{N}}$) Let $\mu : \Sigma^* \to \mathbb{R}_+$ be a function assigning a nonnegative number to each string in Σ^*. We will call μ a *measure* if it satisfies the condition

$$\mu(x) = \sum_{s \in \Sigma} \mu(xs). \tag{2.3.1}$$

We will take the liberty to also write

$$\mu(sX) = \mu(s)$$

for all $s \in \Sigma^*$, and $\mu(\emptyset) = 0$. We call μ a *probability measure* if $\mu(\Lambda) = 1$ and correspondingly, $\mu(X) = 1$.

A measure is called *computable* if it is computable as a function $\mu : \Sigma^* \to \mathbb{R}_+$ according to Definition 1.5.2. ⌟

The following observation lets us extend measures.

Lemma 2.3.6 *Let μ be a measure defined as above. If a cylinder set sX is the disjoint union $xX = x_1X \cup x_2X \cup \ldots$ of cylinder sets, then we have $\mu(z) = \sum_i \mu(x_i)$.*

Proof. Let us call an arbitrary $y \in \Sigma^*$ *good* if

$$\mu(zy) = \sum_i \mu(zyX \cap x_iX)$$

holds. We have to show that the empty string Λ is good.

It is easy to see that y is good if $zyX \subseteq x_iX$ for some i. Also, if ys is good for all $s \in \Sigma$ then y is good. Now assume that Λ is not good. Then there is also an $s_1 \in \Sigma$ that is not good. But then there is also an $s_2 \in \Sigma$ such that s_1s_2 is not good. And so on, we obtain an infinite sequence $s_1s_2\cdots$ such that $s_1 \cdots s_n$ is not good for any n. But then the sequence $zs_1s_2\cdots$ is not contained in $\bigcup_i x_iX$, contrary to the assumption . □

Corollary 2.3.7 *Two disjoint union representations of the same open set*

$$\bigcup_i x_iX = \bigcup_i y_iX$$

imply $\sum_i \mu(x_i) = \sum_i \mu(y_i)$.

Proof. The set $\bigcup_i x_iX$ can also be written as

$$\bigcup_i x_iX = \bigcup_{i,j} x_iX \cap y_jX.$$

An element $x_iX \cap y_jX$ is nonempty only if one of the strings x_i, y_j is a continuation of the other. Calling this string z_{ij} we have $x_iX \cap y_jX = z_{ij}X$. Now Lemma 2.3.6 is applicable to the union $x_iX = \bigcup_j (x_iX \cap y_jX)$, giving

$$\mu(x_i) = \sum_j \mu(x_iX \cap y_jX),$$
$$\sum_i \mu(x_i) = \sum_{i,j} \mu(x_iX \cap y_jX).$$

The right-hand side is also equal similarly to $\sum_j \mu(y_j)$. □

The above corollary allows the following definition:

Definition 2.3.8 For an open set G given as a disjoint union of cylinder sets x_iX let $\mu(G) = \sum_i \mu(x_i)$. For a closed set $F = X \setminus G$ where G is open, let $\mu(F) = \mu(X) - \mu(G)$. ⌟

The following is easy to check.

Proposition 2.3.9 *The measure as defined on open sets is monotonic and countably additive. In particular, it is countably additive on the algebra of finitely determined sets.*

Before extending measures to a wider range of sets, consider an important special case. With infinite sequences, there are certain events that are not impossible, but still have probability 0.

Definition 2.3.10 Given a measure μ, a set $N \subseteq X$ is called a *null set* with respect to μ if there is a sequence $G_1, G_2, \ldots$ of open sets with the property $N \subseteq \bigcap_m G_m$, and $\mu(G_m) \leq 2^{-m}$. We will say that the sequence G_m *witnesses* the fact that N is a null set. ⌟

Examples 2.3.11 Let $\Sigma = \{0, 1\}$, let us define the measure λ by $\lambda(x) = 2^{-n}$ for all n and for all $x \in \Sigma^n$.

1. For every infinite sequence $\xi \in X$, the one-element set $\{\xi\}$ is a null set with respect to λ. Indeed, for each natural number n, let $H_n = \{\xi \in X : \xi(0) = \xi(0), \xi(1) = \xi(1), \ldots, \xi(n) = \xi(n)\}$. Then $\lambda(H_n) = 2^{-n-1}$, and $\{s\} = \bigcap_n H_n$.
2. For a less trivial example, consider the set E of those elements $t \in X$ that are 1 in each positive even position, that is

$$E = \{t \in X : t(2) = 1, t(4) = 1, t(6) = 1 \ldots\}.$$

Then E is a null set. Indeed, for each natural number n, let $G_n = \{t \in X : t(2) = 1, t(4) = 1, \ldots, t(2n) = 1\}$. This helps expressing E as $E = \bigcap_n G_n$, where $\lambda(G_n) = 2^{-n}$.

⌟

Proposition 2.3.12 *Let $N_1, N_2, \ldots$ be a sequence of null sets with respect to a measure μ. Their union $N = \bigcup_i N_i$ is also a null set.*

Proof. Let $G_{i,1}, G_{i,2}, \ldots$ be the infinite sequence witnessing the fact that N_i is a null set. Let $H_m = \bigcup_{i=1}^{\infty} G_{i,m+i}$. Then the sequence H_m of open sets witnesses the fact that N is a null set. □

We would like to extend our measure to null sets N and say $\mu(N) = 0$. The proposition we have just proved shows that such an extension would be countably additive on the null sets. But we do not need this extension for the moment, so we postpone it. Still, given a probability measure P over X, it becomes meaningful to say that a certain property holds with probability 1. What this means is that the set of those sequences that do not have this property is a null set.

Following the idea of Martin-Löf, we would like to call a sequence nonrandom, if it is contained in some *simple* null set; that is it has some easily definable property with probability 0. As we have seen in part 1 of Example 2.3.11, it is important to insist on simplicity, otherwise (say with respect to the measure λ) every sequence might be contained in a null set, namely the one-element set consisting of itself. But most of these sets are not defined simply at all. An example of a simply defined null set is given is part 2 of Example 2.3.11. These reflections justify the following definition, in which "simple" is specified as "constructive".

Definition 2.3.13 Let μ be a computable measure. A set $N \subseteq X$ is called a *constructive null set* if there is recursively enumerable set $\Gamma \subseteq \mathbb{N} \times \Sigma^*$ with the property that denoting $\Gamma_m = \{x : (m, x) \in \Gamma\}$ and $G_m = \bigcup_{x \in \Gamma_m} xX$ we have $N \subseteq \bigcap_m G_m$, and $\mu(G_m) \leq 2^{-m}$. ⌟

In words, the difference between the definition of null sets and constructive null sets is that the sets $G_m = \bigcup_{x \in \Gamma_m} xX$ here are required to be constructive open, moreover, in such a way that from m one can compute the program generating G_m. In even looser words, a set is a constructive null set if for any $\varepsilon > 0$ one can construct effectively a union of cylinder sets containing it, with total measure $\leq \varepsilon$.

Now we are in a position to define random infinite sequences.

Definition 2.3.14 (Random sequence) An infinite sequence ξ is *random* with respect to a probability measure P if and only if ξ is not contained in any constructive null set with respect to P. ⌟

It is easy to see that the set of nonrandom sequences is a null set. Indeed, there is only a countable number or constructive null sets, so even their union is a null set. The following theorem strengthens this observation significantly.

Theorem 2.3.1 *Let us fix a* computable *probability measure P. The set of all nonrandom sequences is a constructive null set.*

Thus, there is a *universal* constructive null set, containing all other constructive null sets. A sequence is random when it does not belong to this set.

Proof of Theorem 2.3.1. The proof uses the projection technique that has appeared several times in this book, for example in proving the existence of a universal lower semicomputable semimeasure.

We know that it is possible to list all recursively enumerable subsets of the set $\mathbb{N} \times \Sigma^*$ into a sequence, namely that there is a recursively enumerable set $\Delta \subseteq \mathbb{N}^2 \times \Sigma^*$ with the property that for every recursively enumerable set $\Gamma \subseteq \mathbb{N} \times \Sigma^*$ there is an e with $\Gamma = \{(m, x) : (e, m, x) \in \Delta\}$. We will write

$$\Delta_{e,m} = \{x : (e, m, x) \in \Delta\},$$
$$D_{e,m} = \bigcup_{x \in \Delta_{e,m}} xX.$$

We transform the set Δ into another set Δ' with the following property.

- For each e, m we have $\sum_{(e,m,x)\in\Delta'} \mu(x) \leq 2^{-m+1}$.
- If for some e we have $\sum_{(e,m,x)\in\Delta} \mu(x) \leq 2^{-m}$ for all m then for all m we have $\{x : (e, m, x) \in \Delta'\} = \{x : (e, m, x) \in \Delta\}$.

This transformation is routine, so we leave it to the reader. By the construction of Δ', for every constructive null set N there is an e with

$$N \subseteq \bigcap_m D'_{e,m}.$$

Define the recursively enumerable set

$$\hat{\Gamma} = \{(m, x) : \exists e \, (e, m + e + 2, x) \in \Delta'\}.$$

Then $\hat{G}_m = \bigcup_e D'_{e,m+e+2}$. For all m we have

$$\sum_{x \in \hat{\Gamma}_m} \mu(x) = \sum_e \sum_{(e,m+e+2,x)\in\Delta'} \mu(x) \leq \sum_e 2^{-m-e-1} = 2^{-m}.$$

This shows that $\hat{\Gamma}$ defines a constructive null set. Let Γ be any other recursively enumerable subset of $\mathbb{N} \times \Sigma^*$ that defines a constructive null set. Then there is an e such that for all m we have $G_m = D'_{e,m}$. The universality follows now from

$$\bigcap_m G_m = \bigcap_m D'_{e,m} \subseteq \bigcap_m D'_{e,m+e+2} \subseteq \bigcap_m \hat{G}_m.$$

□

2.3.2 Probability space

Now we are ready to extend measure to a much wider range of subsets of X.

Definition 2.3.15 Elements of the smallest σ-algebra $\mathcal{A}$ containing the cylinder sets of X are called the *Borel* sets of X. The pair $(X, \mathcal{A})$ is an example of a *measureable space*, where the Borel sets are called the *measureable sets*.

A nonnegative sigma-additive function μ over $\mathcal{A}$ is called a *measure*. It is called a *probability measure* if $\mu(X) = 1$. If μ is fixed then the triple $(X, \mathcal{A}, \mu)$ is called a *measure space*. If it is a probability measure then the space is called a *probability space*. ⌟

In the Appendix we cited a central theorem of measure theory, Caratheodory's extension theorem. It implies that if a measure is defined on an algebra $\mathcal{L}$ in a sigma-additive way then it can be extended uniquely to the σ-algebra generated by $\mathcal{L}$, that is the smallest σ-algebra containing $\mathcal{L}$. We defined a measure μ over X as a nonnegative function $\mu : \Sigma^* \to \mathbb{R}_+$ satisfying the equality (2.3.1). Then we defined $\mu(xX) = \mu(x)$, and further extended μ in a σ-additive way to all elements of the algebra $\mathcal{F}$. Now Caratheodory's theorem allows us to extend it uniquely to all Borel sets, and thus to define a measureable space $(X, \mathcal{A}, \mu)$.

Of course, all null sets in $\mathcal{A}$ get measure 0.

Open, closed and measureable sets can also be defined in the set of real numbers.

Definition 2.3.16 A subset $G \subseteq \mathbb{R}$ is *open* if it is the union of a set of open intervals (a_i, b_i). It is *closed* if its complement is open.

The set $\mathcal{B}$ of *Borel sets* of $\mathbb{B}$ is defined as the smallest σ-algebra containing all open sets. ⌟

The pair $(\mathbb{R}, \mathcal{B})$ is another example of a measureable space.

Definition 2.3.17 (Lebesgue measure) Consider the set left-closed intervals of the line (including intervals of the form $(-\infty, a)$. Let $\mathcal{L}$ be the set of finite disjoint unions of such intervals. This is an algebra. We define the function λ over $\mathcal{L}$ as follows: $\lambda(\bigcup_i [a_i, b_i)) = \sum_i b_i - a_i$. It is easy to check that this is a σ-additive measure and therefore by Caratheodory's theorem can be extended to the set $\mathcal{B}$ of all Borel sets. This function is called the *Lebesgue measure* over $\mathbb{R}$, giving us the measureable space $(\mathbb{R}, \mathcal{B}, \lambda)$. ⌟

Finally, we can define the notion of a measureable function over X.

Definition 2.3.18 (Measureable functions) A function $f : X \to \mathbb{R}$ is called *measureable* if and only if $f^{-1}(E) \in \mathcal{A}$ for all $E \in \mathcal{B}$. ⌟

The following is easy to prove.

Proposition 2.3.19 *Function $f : X \to \mathbb{R}$ is measureable if and only if all sets of the form $f^{-1}((r, \infty)) = \{x : f(x) > r\}$ are measureable, where r is a rational number.*

2.3.3 Computability

Measureable functions are quite general; it is worth introducing some more resticted kinds of function over the set of sequences.

Definition 2.3.20 A function $f : X \to \mathbb{R}$ is *continuous* if and only if for every $x \in X$, for every $\varepsilon > 0$ there is a cylinder set $C \ni x$ such that $|f(y) - f(x)| < \varepsilon$ for all $y \in C$.

A function $f : X \to \mathbb{R}$ is *lower semicontinuous* if for every $r \in \mathbb{R}$ the set $\{x \in X : f(x) > r\}$ is open. It is *upper semicontinuous* if $-f$ is lower semicontinuous. ⌟

The following is easy to verify.

Proposition 2.3.21 *A function $f : X \to \mathbb{R}$ is continuous if and only if it is both upper and lower semicontinuous.*

Definition 2.3.22 A function $f : X \to \mathbb{R}$ is *computable* if for every open rational interval (u, v) the set $f^{-1}((u, v))$ is a constructive open set of X, uniformly in u, v. ⌟

Informally this means that if for the infinite sequence $\xi \in X$ we have $u < f(\xi) < v$ then from u, v sooner or later we will find a prefix x of ξ with the property $u < f(xX) < v$.

Definition 2.3.23 A function $f : X \to \mathbb{R}$ is *lower semicomputable* if for every rational r the set $\{s \in X : f(s) > r\}$ is a constructive open set of X, uniformly in r. It is *upper semicomputable* if $-f$ is lower semicomputable. ⌟

The following is easy to verify.

Proposition 2.3.24 *A function $X \to \mathbb{R}$ is computable if and only if it is both lower and upper semicomputable.*

2.3.4 Integral

The definition of integral over a measure space is given in the Appendix, in Subsection A.2.3. Here, we give an exposition specialized to infinite sequences.

Definition 2.3.25 A measurable function $f : X \to \mathbb{R}$ is called a *step function* if its range is finite. The set of step functions will be called $\mathcal{E}$.

Given a step function f which takes values x_i on sets A_i, and a finite measure μ, we define

$$\mu(f) = \mu f = \int f \, d\mu = \int f(x)\mu(dx) = \sum_i x_i \, \mu(A_i).$$

⌟

Proposition A.2.14, when specified to our situation here, says the following.

Proposition 2.3.26 *The functional μ defined above on step functions can be extended to the set $\mathcal{E}_+$ of monotonic limits of nonnegative elements of $\mathcal{E}$, by continuity. The set $\mathcal{E}_+$ is the set of all nonnegative measurable functions.*

Now we extend the notion of integral to a wider class of functions.

Definition 2.3.27 A measurable function f is called *integrable* with respect to a finite measure μ if $\mu|f|^+ < \infty$ and $\mu|f|^- < \infty$. In this case, we define $\mu f = \mu|f|^+ - \mu|f|^-$. ⌟

It is easy to see that the mapping $f \mapsto \mu f$ is linear when f runs through the set of measureable functions with $\mu|f| < \infty$.

The following is also easy to check.

Proposition 2.3.28 *Let μ be a computable measure.*

a) *If f is computable then a program to compute μf can be found from the program to compute f.*

b) *If f is lower semicomputable then a program to lower semicompute μf can be found from a program to lower semicompute f.*

2.3.5 Randomness tests

We can now define randomness tests similarly to Section 2.2.

Definition 2.3.29 A function $d : X \to \mathbb{R}$ is an *integrable test*, or *expectation-bounded test* of randomness with respect to the probability measure P if it is lower semicomputable and satisfies the condition

$$\int 2^{d(x)} P(dx) \leq 1. \tag{2.3.2}$$

It is called a *Martin-Löf test*, or *probability-bounded test*, if instead of the latter condition only the weaker one is satisfied saying $P(d(x) > m) < 2^{-m}$ for each positive integer m.

It is *universal* if it dominates all other integrable tests to within an additive constant. Universal Martin-Löf tests are defined in the same way. ⌟

Randomness tests and constructive null sets are closely related.

Theorem 2.3.2 *Let P be a computable probability measure over the set $S = \Sigma^\infty$. There is a correspondence between constructive null sets and randomness tests:*

a) *For every randomness tests d, the set $N = \{\xi : d(\xi) = \infty\}$ is a constructive null set.*

b) *For every constructive null set N there is a randomness test d with $N = \{\xi : d(\xi) = \infty\}$.*

Proof. Let d be a randomness test: then for each k the set $G_k = \{\xi : d(\xi) > k\}$ is a constructive open set, and by Markov's inequality (wich is proved in the continuous case just as in the discrete case) we have $P(G_k) \leq 2^{-k}$. The sets G_k witness that N is a constructive null set.

Let N be a constructive null set with $N \subseteq \bigcap_{k=1}^\infty G_k$, where G_k is a uniform sequence of constructive open sets with $P(G_k) = 2^{-k}$. Without loss of generality assume that the sequence G_k is decreasing. Then the function $d(\xi) = \sup\{k : \xi \in G_k\}$ is lower semicomputable and satisfies $P\{\xi : d(\xi) \geq k\} \leq 2^{-k}$, so it is a Martin-Löf test. Just as in Proposition 2.2.3, it is easy to check that $d(x) - 2\log d(x) - c$ is an integrable test for some constant c. □

Just as there is a universal constructive null set, there are universal randomness tests.

Theorem 2.3.3 *For a computable measure P, there is a universal integrable test $\overline{\mathbf{d}}_P(\xi)$ for any fixed computable probability distribution P. More exactly, the function $\xi \mapsto \overline{\mathbf{d}}_P(\xi)$ is lower semicomputable, satisfies (2.2.1) and for all integrable tests $d(\xi)$ for P, we have*

$$d(\xi) \stackrel{+}{<} \overline{\mathbf{d}}_P(\xi) + H(d) + H(P).$$

The proof is similar to the proof Theorem 1.6.3 on the existence of a universal constructive semimeasure. It uses the projection technique and a weighted sum.

2.3.6 Randomness and complexity

We hope for a result connecting complexity with randomness similar to Theorem 2.2.1. Somewhat surprisingly, there is indeed such a result. This is a surprise since in an infinite sequence, arbitrarily large oscillations of any quantity depending on the prefixes are actually to be expected.

Theorem 2.3.4 *Let $X = \Sigma^{\mathbb{N}}$ be the set of infinite sequences. For all computable measures μ over X, we have*

$$\overline{\mathbf{d}}_\mu(\xi) \stackrel{+}{=} \sup_n (-\log \mu(\xi_{1:n}) - H(\xi_{1:n})). \tag{2.3.3}$$

Here, the constant in $\stackrel{+}{=}$ depends on the computable measure μ.

Corollary 2.3.30 *Let λ be the uniform distribution over the set of infinite binary sequences. Then*

$$\overline{\mathbf{d}}_\lambda(\xi) \stackrel{+}{=} \sup_n (n - H(\xi_{1:n})). \tag{2.3.4}$$

In other words, an infinite binary sequence is random (with respect to the uniform distribution) if and only if the complexity $H(\xi_{1:n})$ of its initial segments of length n never decreases below n by more than an additive constant.

Proof. To prove $\stackrel{+}{<}$, define the function

$$f_\mu(\xi) = \sum_s 1_{sX}(\xi)\frac{\mathbf{m}(s \mid \mu)}{\mu(s)} = \sum_n \frac{\mathbf{m}(\xi_{1:n} \mid \mu)}{\mu(\xi_{1:n})} \geq \sup_n \frac{\mathbf{m}(\xi_{1:n} \mid \mu)}{\mu(\xi_{1:n})}.$$

The function $\xi \mapsto f_\mu(\xi)$ is lower semicomputable with $\mu^\xi f_\mu(\xi) \leq 1$, and hence

$$\overline{\mathbf{d}}_\mu(\xi) \stackrel{+}{>} \log f(\xi) \stackrel{+}{>} \sup_n (-\log \mu(\xi_{1:n}) - H(\xi_{1:n} \mid \mu)).$$

The proof of $\stackrel{+}{<}$ reproduces the proof of Theorem 5.2 of [18]. Let us abbreviate:

$$1_y(\xi) = 1_{yX}(\xi).$$

Since $\lceil d(\xi) \rceil$ only takes integer values and is lower semicomputable, there are computable sequences $y_i \in \Sigma^*$ and $k_i \in \mathbb{N}$ with

$$2^{\lceil d(\xi) \rceil} = \sup_i \; 2^{k_i} 1_{y_i}(\xi) \geq \frac{1}{2} \sum_i 2^{k_i} 1_{y_i}(\xi)$$

with the property that if $i < j$ and $1_{y_i}(\xi) = 1_{y_j}(\xi) = 1$ then $k_i < k_j$. The inequality follows from the fact that for any finite sequence $n_1 < n_2 < \ldots$, $\sum_j 2^{n_j} \leq 2 \max_j 2^{n_j}$. The function $\gamma(y) = \sum_{y_i = y} 2^{k_i}$ is lower semicomputable. With it, we have

$$\sum_i 2^{k_i} 1_{y_i}(\xi) = \sum_{y \in \Sigma^*} 1_y(\xi)\gamma(y).$$

Since $\mu 2^{\lceil d \rceil} \leq 2$, we have $\sum_y \mu(y)\gamma(y) \leq 2$, hence $\mu(y)\gamma(y) \stackrel{*}{<} \mathbf{m}(y)$, that is $\gamma(y) \stackrel{*}{<} \mathbf{m}(y)/\mu(y)$. It follows that

$$2^{d(\xi)} \stackrel{*}{<} \sup_{y \in \Sigma^*} \; 1_y(\xi)\frac{\mathbf{m}(y)}{\mu(y)} = \sup_n \frac{\mathbf{m}(\xi_{1:n})}{\mu(\xi_{1:n})}.$$

Taking logarithms, we obtain the $\stackrel{+}{<}$ part of the theorem. □

2.3.7 Universal semimeasure, algorithmic probability

Universal semimeasures exist over infinite sequences just as in the discrete space.

Definition 2.3.31 A function $\mu : \Sigma^* \to \mathcal{R}_+$ is a *semimeasure* if it satisfies

$$\mu(x) \geq \sum_{s \in \Sigma} \mu(xs),$$
$$\mu(\Lambda) \leq 1.$$

If it is lower semicomputable we will call the semimeasure *constructive*. ⌟

These requirements imply, just as for measures, the following generalization to an arbitrary prefix-free set $A \subseteq \Sigma^*$:

$$\sum_{x \in A} \mu(x) \leq 1.$$

Standard technique gives the following:

Theorem 2.3.5 (Universal semimeasure) *There is a universal constructive semimeasure μ, that is a constructive semimeasure with the property that for every other constructive semimeasure ν there is a constant $c_\nu > 0$ with $\mu(x) \geq c_\nu \nu(x)$.*

Definition 2.3.32 Let us fix a universal constructive semimeasure over X and denote it by $M(x)$. ⌟

There is a graphical way to represent a universal semimeasure, via monotonic Turing machines, which can be viewed a generalization of self-delimiting machines.

Definition 2.3.33 A Turing machine $\mathcal{T}$ is *monotonic* if it has no input tape or output tape, but can ask repeatedly for input symbols and can emit repeatedly output symbols. The input is assumed to be an infinite binary string, but it is not assumed that $\mathcal{T}$ will ask for all of its symbols, even in infinite time. If the finite or infinite input string is $p = p_1 p_2 \cdots$, the (finite or infinite) output string is written as $T(p) \in \Sigma^* \cup \Sigma^{\mathbb{N}}$. ⌟

Just as we can generalize self-delimiting machines to obtain the notion of monotonic machines, we can generalize algorithmic probability, defined for a discrete space, to a version defined for infinite sequences. Imagine the monotonic Turing machine $\mathcal{T}$ computing the function $T(p)$ as in the definition, and assume that its input symbols come from coin-tossing.

Definition 2.3.34 (Algorithmic probability over infinite sequences) For a string $x \in \Sigma^*$, let $P_T(x)$ be the probability of $T(\pi) \sqsupseteq x$, when $\pi = \pi_1 \pi_2 \cdots$ is the infinite coin-tossing sequence. ⌟

We can compute

$$P_T(x) = \sum_{T(p)\sqsupseteq x, p \text{ minimal}} 2^{-l(p)},$$

where "minimal" means that no prefix p' of p gives $T(p') \sqsupseteq x$. This expression shows that $P_T(x)$ is lower semicomputable. It is also easy to check that it is a semimeasure, so we have a constructive semimeasure. The following theorem is obtained using a standard construction:

Theorem 2.3.6 *Every constructive semimeasure $\mu(x)$ can be represented as $\mu(x) = P_T(x)$ with the help of an appropriate a monotonic machine $\mathcal{T}$.*

This theorem justifies the following.

Definition 2.3.35 From now on we will call the universal semimeasure $M(x)$ also the *algorithmic probability*. A monotonic Turing machine $\mathcal{T}$ giving $P_T(x) = M(x)$ will be called an *optimal machine*. ⌟

We will see that $-\log M(x)$ should also be considered a kind of description complexity:

Definition 2.3.36 Denote

$$KM(x) = -\log M(x).$$

⌟

How does $KM(x)$ compare to $H(x)$?

Proposition 2.3.37 *We have the bounds*

$$KM(x) \stackrel{+}{<} H(x) \stackrel{+}{<} KM(x) + H(l(x)).$$

Proof. To prove $KM(x) \stackrel{+}{<} H(x)$ define the lower semicomputable function

$$\mu(x) = \sup_S \sum_{y\in S} \mathbf{m}(xy),$$

By its definition this is a constructive semimeasure and it is $\geq \mathbf{m}(x)$. This implies $M(x) \stackrel{*}{>} \mathbf{m}(x)$, and thus $KM(x) \stackrel{+}{<} H(x)$.

For the other direction, define the following lower semicomputable function ν over Σ^*:

$$\nu(x) = \mathbf{m}(l(x)) \cdot M(x).$$

To show that it is a semimeasure compute:

$$\sum_{x\in\Sigma^*} \nu(x) = \sum_n \mathbf{m}(n) \sum_{x\in\Sigma^n} M(x) \leq \sum_n \mathbf{m}(n) \leq 1.$$

It follows that $\mathbf{m}(x) \stackrel{*}{>} \nu(x)$ and hence $H(x) \stackrel{+}{<} KM(x) + H(l(x))$. □

2.3.8 Randomness via algorithmic probability

Fix a computable measure $P(x)$ over the space X. Just as in the discrete case, the universal semimeasure gives rise to a Martin-Löf test.

Definition 2.3.38 Denote

$$\mathbf{d}'_P(\xi) = \log \sup_n \frac{M(\xi_{1:n})}{P(\xi_{1:n})}.$$

⌟

By Proposition 2.3.37, this function is is $\stackrel{*}{>} \overline{\mathbf{d}}_P(\xi)$, defined in Theorem 2.3.3. It can be shown that it is no longer expectation-bounded. But the theorem below shows that it is still a Martin-Löf (probability-bounded) test, so it defines the same infinite random sequences.

Theorem 2.3.7 *The function $\mathbf{d}'_P(\xi)$ is a Martin-Löf test.*

Proof. Lower semicomputability follows from the form of definition. By standard methods, produce a list of finite strings $y_1, y_2, \ldots$ with:

a) $M(y_i)/P(y_i) > 2^m$

b) The cylinder sets $y_i X$ cover of G_m, and are disjoint (thus the set $\{y_1, y_2, \ldots\}$ is prefix-free).

We have

$$P(G_m) = \sum_i P(y_i) < 2^{-m} \sum_i M(y_i) \leq 2^{-m}.$$

□

Just as in the discrete case, the universality of $M(x)$ implies

$$KM(x) \stackrel{+}{<} -\log P(x) + H(P).$$

On the other hand,

$$\sup_n -\log P(\xi_{1:n}) - KM(\xi_{1:n})$$

is a randomness test. So for random ξ, the value $KM(\xi_{1:n})$ remains within a constant of $-\log P(x)$: it does not oscillate much.

3 Information

3.1 Information-theoretic relations

In this subsection, we use some material from [23].

3.1.1 The information-theoretic identity

Information-theoretic entropy is related to complexity in both a formal and a more substantial way.

Classical entropy and classical coding

Entropy is used in information theory in order to characterize the compressibility of a statistical source.

Definition 3.1.1 The *entropy* of a discrete probability distribution P is defined as

$$\mathcal{H}(P) = -\sum_x P(x) \log P(x).$$

⌟

A simple argument shows that this quantity is non-negative. It is instructive to recall a more general inequality, taken from [13], implying it:

Theorem 3.1.1 *Let $a_i, b_i > 0$, $a = \sum_i a_i$, $b = \sum_i b_i$. Then we have*

$$\sum_i a_i \log \frac{a_i}{b_i} \geq a \log \frac{a}{b}, \tag{3.1.1}$$

with equality only if a_i/b_i is constant. In particular, if $\sum_i a_i = 1$ and $\sum_i b_i \leq 1$ then we have $\sum_i a_i \log \frac{a_i}{b_i} \geq 0$.

Proof. Exercise, an application of Jensen's inequality to the concave function $\log x$. □

If P is a probability distribution and $Q(x) \geq 0$, $\sum_x Q(x) \leq 1$ then this inequality implies

$$H(P) \leq \sum_x P(x) \log Q(x).$$

We can interpret this inequality as follows. We have seen in Subsection 1.6.3 that if $x \mapsto z_x$ is a prefix code, mapping the objects x into binary strings z_x that are not prefixes of each other then $\sum_x 2^{-l(z_x)} \leq 1$. Thus, substituting $Q(x) = 2^{-l(z_x)}$ above we obtain Shannon's result:

Theorem 3.1.2 (Shannon) *If $x \mapsto z_x$ is a prefix code then for its expected codeword length we have the lower bound:*

$$\sum_x P(x) l(z_x) \geq H(P).$$

Entropy as expected complexity

Applying Shannon's theorem 3.1.2 to the code obtained by taking z_x as the shortest self-delimiting description of x, we obtain the inequality

$$\sum_x P(x) H(x) \geq \mathcal{H}(P)$$

On the other hand, since $\mathbf{m}(x)$ is a universal semimeasure, we have $\mathbf{m}(x) \stackrel{*}{>} P(x)$: more precisely, $H(x) \stackrel{+}{<} -\log P(x) + H(P)$, leading to the following theorem.

Theorem 3.1.3 *For a computable distribution P we have*

$$\mathcal{H}(P) \leq \sum_x P(x) H(x) \stackrel{+}{<} \mathcal{H}(P) + H(P). \tag{3.1.2}$$

Note that $\mathcal{H}(P)$ is the entropy of the distribution P while $H(P)$ is just the description complexity of the function $x \mapsto P(x)$, it has nothing to do directly with the magnitudes of the values $P(x)$ for each x as real numbers. These two relations give

$$H(P) \stackrel{+}{=} \sum_x P(x) H(x) - H(P),$$

the entropy is within an additive constant equal to the *expected complexity*. Our intended interpretation of $H(x)$ as information content of the individual object x is thus supported by a tight quantitative relationship to Shannon's statistical concept.

Identities, inequalities

The statistical entropy obeys meaningful identities which immediately follow from the definition of conditional entropy.

Definition 3.1.2 Let X and Y be two discrete random variables with a joint distribution. The *conditional entropy* of Y with respect to X is defined as

$$\mathcal{H}(Y \mid X) = -\sum_x P[X = x] \sum_y P[Y = y \mid X = x] \log P[Y = y \mid X = x].$$

The *joint entropy* of X and Y is defined as the entropy of the pair (X, Y). ⌟

The following *additivity property* is then verifiable by simple calculation:

$$\mathcal{H}(X, Y) = \mathcal{H}(X) + \mathcal{H}(Y \mid X). \tag{3.1.3}$$

Its meaning is that the amount of information in the pair (X, Y) is equal to the amount of information in X plus the amount of our residual uncertainty about Y once the value of X is known.

Though intuitively expected, the following identity needs proof:

$$\mathcal{H}(Y \mid X) \leq \mathcal{H}(Y).$$

We will prove it after we define the difference below.

Definition 3.1.3 The *information* in X about Y is defined by

$$\mathcal{I}(X : Y) = \mathcal{H}(Y) - \mathcal{H}(Y \mid X).$$

⌟

This is the amount by which our knowledge of X decreases our uncertainty about Y.

The identity below comes from equation (3.1.3).

$$\mathcal{I}(X : Y) = \mathcal{I}(Y : X) = \mathcal{H}(X) + \mathcal{H}(Y) - \mathcal{H}(X, Y)$$

The meaning of the symmetry relation is that the *quantity* of information in Y about X is equal to the quantity of information in X about Y. Despite its simple derivation, this fact is not very intuitive; especially since only the quantities are equal, the actual contents are generally not (see [22]). We also call $\mathcal{I}(X : Y)$ the *mutual information* of X and Y. We can also write

$$\mathcal{I}(X : Y) = \sum_{x,y} P[X = x, Y = y] \log \frac{P[X = x, Y = y]}{P[X = x]P[Y = y]}. \tag{3.1.4}$$

Proposition 3.1.4 *The information $\mathcal{I}(X : Y)$ is nonnegative and is 0 only if X, Y are independent.*

Proof. Apply Theorem 3.1.1 to (3.1.4). This is 0 only if $P[X = x, Y = y] = P[X = x]P[Y = y]$ for all x, y. □

The algorithmic addition theorem

For the notion of algorithmic entropy $H(x)$ defined in Section 1.6, it is a serious test of fitness to ask whether it has an additivity property analogous to the identity (3.1.3). Such an identity would express some deeper relationship than the trivial identity (3.1.3) since the conditional complexity is defined using the notion of *conditional computation*, and not by an algebraic formula involving the unconditional definition.

A literal translation of the identity (3.1.3) turns out to be true for the function $H(x)$ as well as for Kolmogorov's complexity $K(x)$ with good approximation. But the exact additivity property of algorithmic entropy is subtler.

Theorem 3.1.4 (Addition) *We have*

$$H(x, y) \stackrel{+}{=} H(x) + H(y \mid x, H(x)). \tag{3.1.5}$$

Proof. We first prove $\stackrel{+}{<}$. Let F be the following s.d. interpreter. The machine computing $F(p)$ tries to parse p into $p = uv$ so that $T(u)$ is defined, then outputs the pair $(T(u), T(v, \langle T(u), l(u)\rangle))$. If u is a shortest description of x and v a shortest description of y given the pair $(x, H(x))$ then the output of F is the pair (x, y).

Now we prove $\stackrel{+}{>}$. Let c be a constant (existing in force of Lemma 1.7.3) such that for all x we have $\sum_y 2^{-H(x,y)} \leq 2^{-H(x)+c}$. Let us define the semicomputable function $f(z, y)$ by

$$f((x, k), y) = 2^{k-H(x,y)-c}.$$

Then for $k = H(x)$ we have $\sum_y f(z, y) \leq 1$, hence the generalization (1.6.13) of the coding theorem implies $H(y \mid x, H(x)) \stackrel{+}{<} H(x, y) - H(x)$. □

Recall the definition of x^* as the first shortest description of x.

Corollary 3.1.5 *We have*

$$H(x^*, y) \stackrel{+}{=} H(x, y). \tag{3.1.6}$$

The Addition Theorem implies

$$H(x, y) \stackrel{+}{<} H(x) + H(y \mid x) \stackrel{+}{<} H(x) + H(y)$$

while these relations do not hold for Kolmogorov's complexity $K(x)$, as pointed out in the discussion of Theorem 1.4.2. For an ordinary interpreter, extra information is needed to separate the descriptions of x and y. The self-delimiting interpreter accepts only a description that provides the neccesary information to determine its end.

The term $H(x)$ cannot be omitted from the condition in $H(y \mid x, H(x))$ in the identity (3.1.5). Indeed, it follows from equation (1.7.2) that with $y = H(x)$, we have $H(x, y) - H(x) \stackrel{+}{=} 0$. But $H(H(x) \mid x)$ can be quite large, as shown in the following theorem given here without proof.

Theorem 3.1.5 (see [17]) *There is a constant c such that for all n there is a binary string x of length n with*

$$H(H(x) \mid x) \geq \log n - \log\log n - c.$$

We will prove this theorem in Section 3.3. For the study of information relations, let us introduce yet another notation.

Definition 3.1.6 For any functions f, g, h with values in $\mathbb{S}$, let

$$f \preceq g \bmod h$$

mean that there is a constant c such that $H(f(x) \mid g(x), h(x)) \leq c$ for all x. We will omit mod$h$ if h is not in the condition. The sign $\asymp$ will mean inequality in both directions. ⌟

The meaning of $x \asymp y$ is that the objects x and y hold essentially the same information. Theorem 1.4.1 implies that if $f \asymp g$ then we can replace f by g wherever it occurs as an argument of the functions H or K. As an illustration of this definition, let us prove the relation

$$x^* \asymp \langle x, H(x)\rangle$$

which implies $H(y \mid x, H(x)) \stackrel{+}{=} H(y \mid x^*)$, and hence

$$H(x, y) - H(x) \stackrel{+}{=} H(y \mid x^*). \tag{3.1.7}$$

The direction $\preceq$ holds because we can compute $x = T(x^*)$ and $H(x) = l(x^*)$ from x^*. On the other hand, if we have x and $H(x)$ then we can enumerate the set $E(x)$ of all shortest descriptions of x. By the corollary of Theorem 1.7.1, the number of elements of this set is bounded by a constant. The estimate (1.6.4) implies $H(x^* \mid x, H(x)) \stackrel{+}{=} 0$.

Though the string x^* and the pair $(x, H(x))$ contain the same information, they are not equivalent under some stronger criteria on the decoder. To obtain x

from the shortest description may take extremely long time. But even if this time t is acceptable, to find any description of length $H(x)$ for x might require about $t2^{H(x)}$ steps of search.

The Addition Theorem can be made formally analogous to its information-theoretic couterpart using Chaitin's notational trick.

Definition 3.1.7 Let $H^*(y \mid x) = H(y \mid x^*)$. ⌟

Of course, $H^*(x) = H(x)$. Now we can formulate the relation (3.1.7) as

$$H^*(x, y) \stackrel{+}{=} H^*(x) + H^*(y \mid x).$$

However, we prefer to use the function $H(y \mid x)$ since the function $H^*(y \mid x)$ is not semicomputable. Indeed, if there was a function $h(x, y)$ upper semicomputable in $\langle x, y\rangle$ such that $H(x, y) - H(x) \stackrel{+}{=} h(x, y)$ then by the argument of the proof of the Addition Theorem, we would have $H(y \mid x) \stackrel{+}{<} h(x, y)$ which is not true, according to remark after the proof of the Addition Theorem.

Definition 3.1.8 We define the algorithmic mutual information with the same formula as classical information:

$$I(x : y) = H(y) - H(y \mid x).$$

⌟

Since the quantity $I(x : y)$ is not quite equal to $H(x) - H(x \mid y)$ (in fact, Levin showed (see [17]) that they are not even asymptotically equal), we prefer to define information also by a symmetrical formula.

Definition 3.1.9 The mutual information of two objects x and y is

$$I^*(x : y) = H(x) + H(y) - H(x, y).$$

⌟

The Addition Theorem implies the identity

$$I^*(x : y) \stackrel{+}{=} I(x^* : y) \stackrel{+}{=} I(y^* : x).$$

Let us show that classical information is indeed the expected value of algorithmic information, for a computable probability distribution p:

Lemma 3.1.10 *Given a computable joint probability mass distribution $p(x, y)$ over (x, y) we have*

$$\begin{aligned} \mathcal{I}(X : Y) - H(p) &\stackrel{+}{<} \sum_x \sum_y p(x, y) I^*(x : y) \\ &\stackrel{+}{<} \mathcal{I}(X : Y) + 2H(p), \end{aligned} \tag{3.1.8}$$

where $H(p)$ is the length of the shortest prefix-free program that computes $p(x, y)$ from input (x, y).

Proof. We have

$$\sum_{x,y} p(x,y) I^*(x:y) \stackrel{+}{=} \sum_{x,y} p(x,y)[H(x) + H(y) - H(x,y)].$$

Define $\sum_y p(x,y) = p_1(x)$ and $\sum_x p(x,y) = p_2(y)$ to obtain

$$\sum_{x,y} p(x,y) I(x:y) \stackrel{+}{=} \sum_x p_1(x) H(x) + \sum_y p_2(y) H(y) - \sum_{x,y} p(x,y) H(x,y).$$

The distributions p_i $(i = 1, 2)$ are computable. We have seen in (3.1.2) that $\mathcal{H}(q) \stackrel{+}{<} \sum_x q(x) H(x) \stackrel{+}{<} \mathcal{H}(q) + H(q)$.

Hence, $\mathcal{H}(p_i) \stackrel{+}{<} \sum_x p_i(x) H(x) \stackrel{+}{<} \mathcal{H}(p_i) + H(p_i)$. $(i = 1, 2)$, and $\mathcal{H}(p) \stackrel{+}{<} \sum_{x,y} p(x,y) H(x,y) \stackrel{+}{<} \mathcal{H}(p) + H(p)$. On the other hand, the probabilistic mutual information is expressed in the entropies by $\mathcal{I}(X:Y) = \mathcal{H}(p_1) + \mathcal{H}(p_2) - \mathcal{H}(p)$. By construction of the q_i's above, we have $H(p_1), H(p_2) \stackrel{+}{<} H(p)$. Since the complexities are positive, substitution establishes the lemma. □

Remark 3.1.11 The information $I^*(x:y)$ can be written as

$$I^*(x:y) \stackrel{+}{=} \log \frac{\mathbf{m}(x,y)}{\mathbf{m}(x)\mathbf{m}(y)}.$$

Formally, $I^*(x:y)$ looks like $\overline{\mathbf{d}}_{\mathbf{m}\times\mathbf{m}}((x,y) \mid \mathbf{m}\times\mathbf{m})$ with the function $\overline{\mathbf{d}}_\cdot(\cdot)$ introduced in (2.2.2). Thus, it looks like $I^*(x:y)$ measures the deficiency of randomness with respect to the distribution $\mathbf{m}\times\mathbf{m}$. The latter distribution expresses our "hypothesis" that x, y are "independent" from each other. There is a serious technical problem with this interpretation: the function $\overline{\mathbf{d}}_P(x)$ was only defined for computable measures P. Though the definition can be extended, it is not clear at all that the expression $\overline{\mathbf{d}}_P(x)$ will also play the role of universal test for arbitrary non-computable distributions. Levin's theory culminating in [32] develops this idea, and we return to it in later parts of these notes. ⌟

Let us examine the size of the defects of naive additivity and information symmetry.

Corollary 3.1.12 *For the defect of additivity, we have*

$$H(x) + H(y \mid x) - H(x,y) \stackrel{+}{=} I(H(x) : y \mid x) \stackrel{+}{<} H(H(x) \mid x). \tag{3.1.9}$$

For information asymmetry, we have

$$I(x:y) - I(y:x) \stackrel{+}{=} I(H(y) : x \mid y) - I(H(x) : y \mid x). \tag{3.1.10}$$

Proof. Immediate from the addition theorem. □

Theorem 3.1.6 (Levin) *The information $I(x : y)$ is not even asymptotically symmetric.*

Proof. For some constant c, assume $|x| = n$ and $H(H(x) \mid x) \geq \log n - \log\log n - c$. Consider x, x^* and $H(x)$. Then we can check immediately the relations

$$I(x : H(x)) \stackrel{+}{<} 3\log\log n < \log n - \log\log n - c \stackrel{+}{<} I(x^* : H(x)).$$

and

$$I(H(x) : x) \stackrel{+}{=} I(H(x) : x^*).$$

It follows that symmetry breaks down in an exponentially great measure either on the pair $(x, H(x))$ or the pair $(x^*, H(x))$. □

Data processing

The following useful identity is also classical, and is called the *data processing identity*:

$$\mathcal{I}(Z : (X, Y)) = \mathcal{I}(Z : X) + \mathcal{I}(Z : Y \mid X). \tag{3.1.11}$$

Let us see what corresponds to this for algorithmic information.

Here is the correct conditional version of the addition theorem:

Theorem 3.1.7 *We have*

$$H(x, y \mid z) \stackrel{+}{=} H(x \mid z) + H(y \mid x, H(x \mid z), z). \tag{3.1.12}$$

Proof. The same as for the unconditional version. □

Remark 3.1.13 The reader may have guessed the inequality

$$H(x, y \mid z) \stackrel{+}{=} H(x \mid z) + H(y \mid x, H(x), z),$$

but it is incorrect: taking $z = x$, $y = H(x)$, the left-hand side equals $H(x^* \mid x)$, and the right-hand side equals $H(x \mid x) + H(H(x) \mid x^*, x) \stackrel{+}{=} 0$. ⌟

The following inequality is a useful tool. It shows that conditional complexity is a kind of one-sided distance, and it is the algorithmic analogue of the well-known (and not difficult classical inequality)

$$\mathcal{H}(X \mid Y) \leq \mathcal{H}(Z \mid Y) + \mathcal{H}(X \mid Z).$$

Theorem 3.1.8 (Simple triangle inequality) *We have*

$$H(z \mid x) \stackrel{+}{<} H(y, z \mid x) \stackrel{+}{<} H(y \mid x) + H(z \mid y). \tag{3.1.13}$$

Proof. According to (3.1.12), we have

$$H(y, z \mid x) \stackrel{+}{=} H(y \mid x) + H(z \mid y, H(y \mid x), x).$$

The left-hand side is $\stackrel{+}{>} H(z \mid x)$, the second term of the right-hand side is $\stackrel{+}{<} H(z \mid y)$. □

Equation (3.1.12) justifies the following.

Definition 3.1.14

$$I^*(x : y \mid z) = H(x \mid z) + H(y \mid z) - H(x, y \mid z). \tag{3.1.14}$$

⌟

Then we have

$$\begin{aligned} I^*(x : y \mid z) &\stackrel{+}{=} H(y \mid z) - H(y \mid x, H(x \mid z), z) \\ &\stackrel{+}{=} H(x \mid z) - H(x \mid y, H(y \mid z), z). \end{aligned}$$

Theorem 3.1.9 (Algorithmic data processing identity) *We have*

$$I^*(z : (x, y)) \stackrel{+}{=} I^*(z : x) + I^*(z : y \mid x^*). \tag{3.1.15}$$

Proof. We have

$$\begin{aligned} I^*(z : (x, y)) &\stackrel{+}{=} H(x, y) + H(z) - H(x, y, z) \\ I^*(z : x) &\stackrel{+}{=} H(x) + H(z) - H(x, z) \\ I^*(z : (x, y)) - I^*(z : x) &\stackrel{+}{=} H(x, y) + H(x, z) - H(x, y, z) - H(x) \\ &\stackrel{+}{=} H(y \mid x^*) + H(z \mid x^*) - H(y, z \mid x^*) \\ &\stackrel{+}{=} I^*(z : y \mid x^*), \end{aligned}$$

where we used additivity and the definition (3.1.14) repeatedly. □

Corollary 3.1.15 *The information I^* is monotonic:*

$$I^*(x : (y, z)) \stackrel{+}{>} I^*(x : y). \tag{3.1.16}$$

The following theorem is also sometimes useful.

Theorem 3.1.10 (Starry triangle inequality) *For all x, y, z, we have*

$$H(x \mid y^*) \stackrel{+}{<} H(x, z \mid y^*) \stackrel{+}{<} H(z \mid y^*) + H(x \mid z^*). \tag{3.1.17}$$

Proof. Using the algorithmic data-processing identity (3.1.15) and the definitions, we have

$$\begin{aligned} &H(z) - H(z \mid y^*) + H(x \mid z^*) - H(x \mid y, H(y \mid z^*), z^*) \\ &\quad \stackrel{+}{=} I^*(y : z) + I^*(y : x \mid z^*) \stackrel{+}{=} I^*(y : (x, z)) \\ &\quad \stackrel{+}{=} H(x, z) - H(x, z \mid y^*), \end{aligned}$$

which gives, after cancelling $H(z) + H(x \mid z^*)$ on the left-hand side with $H(x, z)$ on the right-hand side, to which it is equal according to additivity, and changing sign:

$$H(x, z \mid y^*) \stackrel{+}{=} H(z \mid y^*) + H(x \mid y, H(y \mid z^*), z^*).$$

From here, we proceed as in the proof of the simple triangle inequality (3.1.13) □

3.1.2 Information non-increase

In this subsection, we somewhat follow the exposition of [23].

The classical data processing identity has a simple but important application. Suppose that the random variables Z, X, Y form a Markov chain in this order. Then $I(Z : Y \mid X) = 0$ and hence $I(Z : (X, Y)) = I(Z : X)$. In words: all information in Z about Y is coming through X: a Markov transition from X to Y cannot increase the information about Z. Let us try to find the algorithmic counterpart of this theorem.

Here, we rigorously show that this is the case in the algorithmic statistics setting: the information in one object about another cannot be increased by any deterministic algorithmic method by more than a constant. With added randomization this holds with overwhelming probability. For more elaborate versions see [30, 32].

Suppose we want to obtain information about a certain object x. It does not seem to be a good policy to guess blindly. This is confirmed by the following inequality.

$$\sum_y \mathbf{m}(y) 2^{I(x:y)} \stackrel{*}{<} 1 \tag{3.1.18}$$

which says that for each x, the expected value of $2^{I(x:y)}$ is small, even with respect to the universal constructive semimeasure $\mathbf{m}(y)$. The proof is immediate if we recognize that by the Coding Theorem,

$$2^{I(x:y)} \stackrel{*}{=} \frac{2^{-H(y|x)}}{\mathbf{m}(y)}$$

and use the estimate (1.6.10).

We prove a strong version of the information non-increase law under deterministic processing (later we need the attached corollary):

Theorem 3.1.11 *Given x and z, let q be a program computing z from x^*. Then we have*

$$I^*(y : z) \stackrel{+}{<} I^*(y : x) + H(q).$$

Proof. By monotonicity (3.1.16) and the data processing identity (3.1.15):

$$I^*(y : z) \stackrel{+}{<} I^*(y : (x, z)) \stackrel{+}{=} I^*(y : x) + I^*(y : z \mid x^*).$$

By definition of information and the definition of q:

$$I^*(y : z \mid x^*) \stackrel{+}{<} H(z \mid x^*) \stackrel{+}{<} H(q).$$

□

Randomized computation can increase information only with negligible probability. Suppose that z is obtained from x by some randomized computation. The probability $p(z \mid x)$ of obtaining z from x is a semicomputable distribution over the z's. The information increase $I^*(z : y) - I^*(x : y)$ satisfies the theorem below.

Theorem 3.1.12 *For all x, y, z we have*

$$\mathbf{m}(z \mid x^*) 2^{I^*(z:y) - I^*(x:y)} \stackrel{*}{<} \mathbf{m}(z \mid x^*, y, H(y \mid x^*)).$$

Therefore it is upperbounded by $\mathbf{m}(z \mid x) \stackrel{*}{<} \mathbf{m}(z \mid x^*)$.

Remark 3.1.16 The theorem implies

$$\sum_z \mathbf{m}(z \mid x^*) 2^{I^*(z:y) - I^*(x:y)} \stackrel{*}{<} 1. \tag{3.1.19}$$

This says that the $\mathbf{m}(\cdot \mid x^*)$-expectation of the exponential of the increase is bounded by a constant. It follows that for example, the probability of an increase of mutual information by the amount d is $\stackrel{*}{<} 2^{-d}$. ⌟

Proof. We have, by the data processing inequality:

$$I^*(y : (x, z)) - I^*(y : x) \stackrel{+}{=} I^*(y : z \mid x^*) \stackrel{+}{=} H(z \mid x^*) - H(z \mid y, H(y \mid x^*), x^*).$$

Hence, using also $I^*(y : (x, z)) \stackrel{+}{>} I^*(y : z)$ by monotonicity:

$$I^*(y : z) - I^*(y : x) - H(z \mid x^*) \stackrel{+}{<} -H(z \mid y, H(y \mid x^*), x^*).$$

Putting both sides into the exponent gives the statement of the theorem. □

Remark 3.1.17 The theorem on information non-increase and its proof look similar to the theorems on randomness conservation. There is a formal connection, via the observation made in Remark 3.1.11. Due to the difficulties mentioned there, we will explore the connection only later in our notes. ⌟

3.2 The complexity of decidable and enumerable sets

If $\omega = \omega(1)\omega(2)\cdots$ is an infinite binary sequence, then we may be interested in how the complexity of the initial segments

$$\omega(1:n) = \omega(1)\cdots\omega(n)$$

grows with n. We would guess that if ω is "random" then $K(\omega(1:n))$ is approximately n; this question will be studied satisfactorily in later sections. Here, we want to look at some other questions.

Can we determine whether the sequence ω is computable, by just looking at the complexity of its initial segments? It is easy to see that if ω is computable then $K(\omega(1:n)) \stackrel{+}{<} \log n$. But of course, if $K(\omega(1:n)) \stackrel{+}{<} \log n$ then it is not necessarily true yet that ω is computable. Maybe, we should put n into the condition. If ω is computable then $K(\omega(1:n) \mid n) \stackrel{+}{<} 0$. Is it true that $K(\omega(1:n) \mid n) \stackrel{+}{<} 0$ for all n then indeed ω is computable? Yes, but the proof is quite difficult (see either its original, attributed to Albert Meyer in [36], or in [59]). The suspicion arises that when measuring the complexity of starting segments of infinite sequences, neither Kolmogorov complexity nor its prefix version are the most natural choice. Other examples will support the suspicion, so let us introduce, following Loveland's paper [36], the following variant.

Definition 3.2.1 Let us say that a program p *decides* a string $x = (x(1), x(2), \ldots)$ on a Turing machine T up to n if for all $i \in \{1, \ldots, n\}$ we have

$$T(p, i) = x(i).$$

Let us further define the *decision complexity*

$$K_T(x; n) = \min\{l(p) : p \text{ decides } x \text{ on } T \text{ up to } n\}.$$

⌟

As for the Kolmogorov complexity, an invariance theorem holds, and there is an optimal machine T_0 for this complexity. Again, we omit the index T, assuming that such an optimal machine has been fixed.

The differences between $K(x)$, $K(x \mid n)$ and $K(x; n)$ are interesting to explore; intuitively, the important difference is that the program achieving the decision complexity of x does not have to offer precise information about the length of x. We clearly have

$$K(x \mid n) \stackrel{+}{<} K(x; n) \stackrel{+}{<} K(x).$$

Examples that each of these inequalities can be strict, are left to exercises.

Remark 3.2.2 If $\omega(1:n)$ is a segment of an infinite sequence ω then we can write

$$K(\omega; n)$$

instead of $K(\omega(1:n); n)$ without confusion, since there is only one way to understand this expression. ⌟

Decision complexity offers an easy characterization of decidable infinite sequences.

Theorem 3.2.1 *Let $\omega = (\omega(1), \omega(2), \dots)$ be an infinite sequence. Then ω is decidable if and only if*

$$K(\omega; n) \stackrel{+}{<} 0. \tag{3.2.1}$$

Proof. If ω is decidable then (3.2.1) is obviously true. Suppose now that (3.2.1) holds: there is a constant c such that for all n we have $K(\omega(1:n); n) < c$. Then there is a program p of length $\le c$ with the property that for infinitely many n, decides x on the optimal machine T_0 up to n. Then for all i, we have $T_0(p, i) = \omega(i)$, showing that ω is decidable. □

Let us use now the new tool to prove a somewhat more surprising result. Let ω be a 0-1 sequence that is the indicator function of a recursively enumerable set. In other words, the set $\{i : \omega(i) = 1\}$ is recursively enumerable. As we know such sequences are not always decidable, so $K(\omega(1:n); n)$ is not going to be bounded. How fast can it grow? The following theorem gives exact answer, showing that it grows only logarithmically.

Theorem 3.2.2 (Barzdin, see [2])

a) *Let ω be the indicator function of a recursively enumerable set E. Then we have*

$$K(\omega; n) \stackrel{+}{<} \log n.$$

b) *The set E can be chosen such that for all n we have $K(\omega; n) \ge \log n$.*

Proof. Let us prove (a) first. Let n' be the first power of 2 larger than n. Let $k(n) = \sum_{i \le n'} \omega(i)$. For a constant d, let $p = p(n, d)$ be a program that contains an initial segment q of size d followed by a the number $k(n)$ in binary notation, and padded to $\lceil \log n \rceil + d$ with 0's. The program q is self-delimiting, so it sees where $k(n)$ begins.

The machine $T_0(p, i)$ works as follows, under the control of q. From the length of the whole p, it determines n'. It begins to enumerate $\omega(1:n')$ until it found $k(n)$ 1's. Then it knows the whole $\omega(1:n')$, so it outputs $\omega(i)$.

Let us prove now (b). Let us list all possible programs q for the machine T_0 as $q = 1, 2, \dots$. Let $\omega(q) = 1$ if and only if $T_0(q, q) = 0$. The sequence ω is

obviously the indicator function of a recursively enumerable set. To show that ω has the desired property, assume that for some n there is a p with $T_0(p, i) = \omega(i)$ for all $i \leq n$. Then $p > n$ since $\omega(p)$ is defined to be different from $T_0(p, p)$. It follows that $l(p) \geq \log n$. □

Decision complexity has been especially convenient for the above theorem. Neither $K(\omega(1 : n))$ nor $K(\omega(1 : n) \mid n)$ would have been suitable to formulate such a sharp result. To analyze the phenomenon further, we introduce some more concepts.

Definition 3.2.3 Let us denote, for this section, by E the set of those binary strings p on which the optimal prefix machine T halts:

$$\begin{aligned} E^t &= \{p : T(p) \text{ halts in } < t \text{ steps}\}, \\ E &= E^\infty. \end{aligned} \tag{3.2.2}$$

Let

$$\chi = \chi_E \tag{3.2.3}$$

be the infinite sequence that is the indicator function of the set E, when the latter is viewed as a set of numbers. ⌟

It is easy to see that the set E is complete among recursively enumerable sets with respect to many-one reduction. The above theorems show that though it contains an infinite amount of information, this information is not stored in the sequence χ densely at all: there are at most $\log n$ bits of it in the segment $\chi(1 : n)$. There is an infinite sequence, though, in which the same information is stored much more densely: as we will see later, maximally densely.

Definition 3.2.4 (Chaitin's Omega) Let

$$\begin{aligned} \Omega^t &= \sum_{p \in E^t} 2^{-l(p)}, \\ \Omega &= \Omega^\infty. \end{aligned} \tag{3.2.4}$$

⌟

Let $\Omega(1 : n)$ be the sequence of the first n binary digits in the expansion of Ω, and let it also denote the binary number $0.\Omega(1) \cdots \Omega(n)$. Then we have

$$\Omega(1 : n) < \Omega < \Omega(1 : n) + 2^{-n}.$$

Theorem 3.2.3 *The sequences Ω and χ are Turing-equivalent.*

Proof. Let us show first that given Ω as an oracle, a Turing machine can compute χ. Suppose that we want to know for a given string p of length k whether $\chi(p) = 1$ that is whether $T(p)$ halts. Let $t(n)$ be the first t for which $\Omega^t > \Omega(1:n)$. If a program p of length n is not in $E^{t(n)}$ then it is not in E at all, since $2^{-l(p)} = 2^n < \Omega - \Omega^t$. It follows that $\chi(1:2^n)$ can be computed from $\Omega(1:n)$.

To show that Ω can be computed from E, let us define the recursively enumerable set

$$E' = \{r : r \text{ rational}, \Omega > r\}.$$

The set E' is reducible to E since the latter is complete among recursively enumerable sets with respect to many-one reduction. On the other hand, Ω is obviously computable from E'. □

The following theorem shows that Ω stores the information more densely.

Theorem 3.2.4 *We have* $H(\Omega(1:n)) \stackrel{+}{>} n$.

Proof. Let p_1 be a self-delimiting program outputting $\Omega(1:n)$. Recall the definition of the sets E^t in (3.2.2) and the numbers Ω^t in (3.2.4). Let Ω_1 denote the real number whose binary digits after 0 are given by $\Omega(1:n)$, and let t_1 be the first t with $\Omega^t > \Omega_1$. Let x_1 be the first string x such that $T(p) \neq x$ for any $p \in E^{t_1}$.

We have $H(x_1) \geq n$. On the other hand, we just computed x_1 from p_1, so $H(x_1 \mid p_1) \stackrel{+}{<} 0$. We found $n \leq H(x_1) \stackrel{+}{<} H(p_1) + H(x_1 \mid p_1) \stackrel{+}{<} H(p_1)$. □

3.3 The complexity of complexity

3.3.1 Complexity is sometimes complex

This section is devoted to a quantitative estimation of the uncomputability of the complexity function $K(x)$. Actually, we will work with the prefix complexity $H(x)$, but the results would be very similar for $K(x)$. The first result shows that the value $H(x)$ is not only not computable from x, but its conditional complexity $H(H(x) \mid x)$ given x is sometimes quite large. How large can it be expected to be? Certainly not much larger than $\log H(x) + 2 \log\log H(x)$, since we can describe any value $H(x)$ using this many bits. But it can come close to this, as shown by Theorem 3.1.5. This theorem says that for all n, there exists x of length n with

$$H(H(x) \mid x) \stackrel{+}{>} \log n - \log\log n. \tag{3.3.1}$$

Proof of Theorem 3.1.5. Let $U(p, x)$ be the optimal self-delimiting machine for which $H(y \mid x) = \min_{U(p,x)=y} l(p)$. Let $s \leq \log n$ be such that if $l(x) = n$ then a p of length x can be found for which $U(p, x) = H(x)$. We will show

$$s \stackrel{+}{>} \log n - \log\log n. \tag{3.3.2}$$

Let us say that $p \in \{0,1\}^s$ is *suitable* for $x \in \{0,1\}^n$ if there exists a $k = U(p,x)$ and a $q \in \{0,1\}^k$ with $U(p,\Lambda) = x$. Thus, p is suitable for x if it produces the length of some program of x, not necessarily a shortest program.

Let M_i denote the set of those x of length n for which there exist at least i different suitable p of length s. We will examine the sequence

$$\{0,1\}^n = M_0 \supseteq M_1 \supseteq \ldots \supseteq M_j \supseteq M_{j+1} = \emptyset,$$

where $M_j \neq \emptyset$. It is clear that $2^s \geq j$. To lowerbound j, we will show that the sets M_i decrease only slowly:

$$\log |M_i| \stackrel{+}{<} \log |M_{i+1}| + 4 \log n. \tag{3.3.3}$$

We can assume

$$\log |M_i \setminus M_{i+1}| \geq \log |M_i| - 1, \tag{3.3.4}$$

otherwise (3.3.3) is trivial. We will write a program that finds an element x of $M_i \setminus M_{i+1}$ with the property $H(x) \geq \log |M_i| - 1$. The program works as follows.

- It finds i, n with the help of descriptions of length $\log n + 2 \log\log n$, and s with the help of a description of length $2 \log\log n$.
- It finds $|M_{i+1}|$ with the help of a description of length $\log |M_{i+1}| + \log n + 2 \log\log n$.
- From these data, it determines the set M_{i+1}, and then begins to enumerate the set $M_i \setminus M_{i+1}$ as $x_1, x_2, \ldots$. For each of these elements x_r, it knows that there are exactly i programs suitable for x_r, find all those, and find the shortest program for x produced by these. Therefore it can compute $H(x_r)$.
- According to the assumption (3.3.4), there is an x_r with $H(x_r) \geq \log |M_i| - 1$. The program outputs the first such x_r.

The construction of the program shows that its length is $\stackrel{+}{<} \log |M_{i+1}| + 4 \log n$, hence for the x we found

$$\log |M_i| - 1 \leq H(x) \stackrel{+}{<} \log |M_{i+1}| + 4 \log n,$$

which proves (3.3.3). This implies $j \geq n/(4 \log n)$, and hence (3.3.2). □

3.3.2 Complexity is rarely complex

Let

$$f(x) = H(H(x) \mid x).$$

We have seen that $f(x)$ is sometimes large. Here, we will show that the sequences x for which this happens are rare. Recall that we defined χ in (3.2.3) as the indicator sequence of the halting problem.

Theorem 3.3.1 *We have*

$$I(\chi : x) \stackrel{+}{>} f(x) - 2.4 \log f(x).$$

In view of the inequality (3.1.18), this shows that such sequences are rare, even in terms of the universal probability, so they are certainly rare if we measure them by any computable distribution. So, we may even call such sequences "exotic".

In the proof, we start by showing that the sequences in question are rare. Then the theorem will follow when we make use of the fact that $f(x)$ is computable from χ.

We need a lemma about the approximation of one measure by another from below.

Lemma 3.3.1 *For any semimeasure ν and measure $\mu \leq \nu$, let $S_m = \{x : 2^m \mu(x) \leq \nu(x)\}$. Then $\mu(S_m) \leq 2^{-m}$. If $\sum_x (\nu(x) - \mu(x)) < 2^{-n}$ then $\nu(S_1) < 2^{-n+1}$.*

The proof is straightforward.

Let

$$X(n) = \{x : f(x) = n\}.$$

Lemma 3.3.2 *We have*

$$\mathbf{m}(X(n)) \stackrel{*}{<} n^{1.2} 2^{-n}.$$

Proof. Using the definition of E^t in (3.2.2), let

$$\mathbf{m}^t(x) = \sum \{p : T(p) = x \text{ in } < t \text{ steps}\},$$
$$H^t(x) = -\log \mathbf{m}^t(x).$$

Then according to the definition (1.6.14) we have $\mathbf{m}(x) = \mathbf{m}^\infty(x)$, and $H(x) \stackrel{+}{=} H^\infty(x)$. Let

$$t(k) = \min\{t : \Omega(1 : k) < \Omega^t\},$$
$$\mu = \mathbf{m}^{t(k)}.$$

Clearly, $H^{t(k)}(x) \stackrel{+}{>} H(x)$. Let us show that $H^{t(k)}(x)$ is a good approximation for $H(x)$ for most x. Let

$$Y(k) = \{x : H(x) \leq H^{t(k)}(x) - 1\}.$$

By definition, for $x \notin Y(k)$ we have

$$|H^{t(k)}(x) - H(x)| \stackrel{+}{=} 0.$$

On the other hand, applying Lemma 3.3.1 with $\mu = \mathbf{m}^{t(k)}$, $\nu = \mathbf{m}$, we obtain

$$\mathbf{m}(Y(k)) < 2^{-k+1}. \tag{3.3.5}$$

Note that

$$H(H^{t(k)}(x) \mid x, \Omega(1:k)) \stackrel{+}{=} 0,$$

therefore for $x \notin Y(k)$ we have

$$H(H(x) \mid x, \Omega(1:k)) \stackrel{+}{=} 0,$$
$$H(H(x) \mid x) \stackrel{+}{<} k + 1.2 \log k.$$

If $n = k + 1.2 \log k$ then $k \stackrel{+}{=} n - 1.2 \log n$, and hence, if $x \notin Y(n - 1.2 \log n)$ then $H(H(x) \mid x) \stackrel{+}{<} n$. Thus, there is a constant c such that

$$X(n) \subseteq Y(n - 1.2 \log n - c).$$

Using (3.3.5) this gives the statement of the lemma. □

Proof of Theorem 3.3.1. Since $f(x)$ is computable from Ω, the function

$$\nu(x) = \mathbf{m}(x) 2^{f(x)} (f(x))^{-2.4}$$

is computable from Ω. Let us show that it is a semimeasure (within a multiplicative constant). Indeed, using the above lemma:

$$\sum_x \nu(x) = \sum_k \sum_{x \in X(k)} \nu(x) = \sum_k 2^k k^{-2.4} \mathbf{m}(X(k)) = \sum_k k^{-1.2} \stackrel{*}{<} 1.$$

Since $\mathbf{m}(\cdot \mid \Omega)$ is the universal semimeasure relative to Ω we find $\mathbf{m}(x \mid \Omega) \stackrel{*}{>} \nu(x)$, hence

$$H(x \mid \Omega) \stackrel{+}{<} -\log \nu(x) = H(x) - f(x) + 2.4 \log f(x),$$
$$I(\Omega : x) \stackrel{+}{>} f(x) - 2.4 \log f(x).$$

Since Ω is equivalent to χ, the proof is complete. □

4 Generalizations

4.1 Continuous spaces, noncomputable measures

This section starts the consideration of randomness in continuous spaces and randomness with respect to noncomputable measures.

4.1.1 Introduction

The algorithmic theory of randomness is well developed when the underlying space is the set of finite or infinite sequences and the underlying probability distribution is the uniform distribution or a computable distribution. These restrictions seem artificial. Some progress has been made to extend the theory to arbitrary Bernoulli distributions by Martin-Löf in [37], and to arbitrary distributions, by Levin in [29, 31, 32]. The paper [25] by Hertling and Weihrauch also works in general spaces, but it is restricted to computable measures. Similarly, Asarin's thesis [1] defines randomness for sample paths of the Brownian motion: a fixed random process with computable distribution.

The exposition here has been inspired mainly by Levin's early paper [31] (and the much more elaborate [32] that uses different definitions): let us summarize part of the content of [31]. The notion of a constructive topological space $\mathbf{X}$ and the space of measures over $\mathbf{X}$ is introduced. Then the paper defines the notion of a uniform test. Each test is a lower semicomputable function $(\mu, x) \mapsto f_\mu(x)$, satisfying $\int f_\mu(x)\mu(dx) \le 1$ for each measure μ. There are also some additional conditions. The main claims are the following.

a) There is a universal test $\mathbf{t}_\mu(x)$, a test such that for each other test f there is a constant $c > 0$ with $f_\mu(x) \le c \cdot \mathbf{t}_\mu(x)$. The *deficiency of randomness* is defined as $\mathbf{d}_\mu(x) = \log \mathbf{t}_\mu(x)$.

b) The universal test has some strong properties of "randomness conservation": these say, essentially, that a computable mapping or a computable randomized transition does not decrease randomness.

c) There is a measure M with the property that for every outcome x we have $\mathbf{t}_M(x) \leq 1$. In the present work, we will call such measures *neutral*.

d) Semimeasures (semi-additive measures) are introduced and it is shown that there is a lower semicomputable semi-measure that is neutral (let us assume that the M introduced above is lower semicomputable).

e) Mutual information $I(x : y)$ is defined with the help of (an appropriate version of) Kolmogorov complexity, between outcomes x and y. It is shown that $I(x : y)$ is essentially equal to $\mathbf{d}_{M\times M}(x, y)$. This interprets mutual information as a kind of "deficiency of independence".

This impressive theory leaves a number of issues unresolved:

1. The space of outcomes is restricted to be a compact topological space, moreover, a particular compact space: the set of sequences over a finite alphabet (or, implicitly in [32], a compactified infinite alphabet). However, a good deal of modern probability theory happens over spaces that are not even locally compact: for example, in case of the Brownian motion, over the space of continuous functions.
2. The definition of a uniform randomness test includes some conditions (different ones in [31] and in [32]) that seem somewhat arbitrary.
3. No simple expression is known for the general universal test in terms of description complexity. Such expressions are nice to have if they are available.

Here, we intend to carry out as much of Levin's program as seems possible after removing the restrictions. A number of questions remain open, but we feel that they are worth to be at least formulated. A fairly large part of the exposition is devoted to the necessary conceptual machinery. This will also allow to carry further some other initiatives started in the works [37] and [29]: the study of tests that test nonrandomness with respect to a whole class of measures (like the Bernoulli measures).

Constructive analysis has been developed by several authors, converging approximately on the same concepts, We will make use of a simplified version of the theory introduced in [55]. As we have not found a constructive measure theory in the literature fitting our purposes, we will develop this theory here, over (constructive) complete separable metric spaces. This generality is well supported by standard results in measure theoretical probability, and is sufficient for constructivizing a large part of current probability theory.

The appendix recalls some of the needed topology, measure theory, constructive analysis and constructive measure theory. We also make use of the notation introduced there.

4.1.2 Uniform tests

We first define tests of randomness with respect to an arbitrary measure. Recall the definition of lower semicomputable real functions on a computable metric space $\mathbf{X}$.

Definition 4.1.1 Let us be given a computable complete metric space $\mathbf{X} = (X, d, D, \alpha)$. For an arbitrary measure $\mu \in \mathcal{M}(\mathbf{X})$, a *$\mu$-test of randomness* is a μ-lower semicomputable function $f : X \to \overline{\mathbb{R}}_+$ with the property $\mu f \leq 1$. We call an element x *random* with respect to μ if $f(x) < \infty$ for all μ-tests f. But even among random elements, the size of the tests quantifies (non-)randomness.

A *uniform test* of randomness is a lower semicomputable function $f : \mathbf{X} \times \mathcal{M}(\mathbf{X}) \to \overline{\mathbb{R}}_+$, written as $(x, \mu) \mapsto f_\mu(x)$ such that $f_\mu(\cdot)$ is a μ-test for each μ. ⌟

The condition $\mu f \leq 1$ guarantees that the probability of those outcomes whose randomness is $\geq m$ is at most $1/m$. The definition of tests is in the spirit of Martin-Löf's tests. The important difference is in the semicomputability condition: instead of restricting the measure μ to be computable, we require the test to be lower semicomputable also in its argument μ.

The following result implies that every μ-test can be extended to a uniform test.

Theorem 4.1.1 *Let ϕ_e, $e = 1, 2, \ldots$ be an enumeration of all lower semicomputable functions $\mathbf{X} \times \mathbf{Y} \times \mathcal{M}(\mathbf{X}) \to \overline{\mathbb{R}}_+$, where $\mathbf{Y}$ is also a computable metric space, and $s : \mathbf{Y} \times \mathcal{M}(\mathbf{X}) \to \overline{\mathbb{R}}$ a lower semicomputable function. There is a recursive function $e \mapsto e'$ with the property that*

a) *For each e, the function $\phi_{e'}(x, y, \mu)$ is everywhere defined with $\mu^x \phi_{e'}(x, y, \mu) \leq s(y, \mu)$.*

b) *For each e, y, μ, if $\mu^x \phi_e(x, y\mu) < s(y, \mu)$ then $\phi_{e'}(\cdot, y, \mu) = \phi_e(\cdot, y, \mu)$.*

This theorem generalizes a theorem of Hoyrup and Rojas (in allowing a lower semicomputable upper bound $s(y, \mu)$).

Proof. By Proposition B.1.38, we can represent $\phi_e(x, y, \mu)$ as a supremum $\phi_e = \sup_i h_{e,i}$ where $h_{e,i}(x, y, \mu)$ is a computable function of e, i monotonically increasing in i. Similarly, we can represent $s(y, \mu)$ as a supremum $\sup_j s_j(y, \mu)$ where $s_j(y, \mu)$ is a computable function monotonically increasing in j. The integral $\mu^x h_{e,i}(x, y, \mu)$ is computable as a function of (y, μ), in particular it is upper semicomputable.

Define $h'_{e,i,j}(x, y, \mu)$ as $h_{e,i}(x, y, \mu)$ for all j, y, μ with $\mu^x h_{e,i}(x, y, \mu) < s_j(y, \mu)$, and 0 otherwise. Since $s_j(y, \mu)$ is computable this definition makes the function $h'_{e,i,j}(x, y, \mu)$ lower semicomputable. The function $h''_{e,i}(x, y, \mu) = \sup_j h'_{e,i,j}(x, y, \mu)$

is then also lower semicomputable, with $h''_{e,i}(x, y, \mu) \leq h_{e,i}(x, y, \mu)$, and $\mu^x h'_{e,i}(x, y, \mu) \leq s(y, \mu)$. Also, $h''_{e,i}(x, y, \mu)$ is monotonically increasing in i. The function $\phi'_e(x, y, \mu) = \sup_i h''_{e,i}(x, y, \mu)$ is then also lower semicomputable, and by Fubini's theorem we have $\mu^x \phi'_e(x, y, \mu) \leq s(y, \mu)$.

Define $\phi_{e'}(x, y, \mu) = \phi'_e(x, y, \mu)$. Consider any e, y, μ such that $\mu^x \phi_e(x, y, \mu) < s(y, \mu)$ holds. Then for every i there is a j with $\mu^x h_{e,i}(x, y, \mu) < s_j(y, \mu)$, and hence $h'_{e,i,j}(x, y, \mu) = h_{e,i}(x, y, \mu)$. It follows that $h''_{e,i}(x, y, \mu) = h_{e,i}(x, y, \mu)$ for all i and hence $\phi'_e(x, y, \mu) = \phi_e(x, y, \mu)$.

□

Corollary 4.1.2 (Uniform extension) *There is an operation $H_e(x, \mu) \mapsto H_{e'}(x, \mu)$ with the property that $H_{e'}(x, \mu)$ is a uniform test and if $2H_e(\cdot, \mu)$ is a μ-test then $H_{e'}(x, \mu) = H_e(x, \mu)$.*

Proof. In Theorem 4.1.1 set $\phi_e(x, y, \mu) = \frac{1}{2} H_e(x, \mu)$ with $s(y, \mu) = 1$. □

Corollary 4.1.3 (Universal generalized test) *Let $s : \mathbf{Y} \times \mathcal{M}(\mathbf{X}) \to \overline{\mathbb{R}}_+$ a lower semicomputable function. Let E be the set of lower semicomputable functions $\phi(x, y, \mu) \geq 0$ with $\mu^x \phi(x, y, \mu) \leq s(y, \mu)$. There is a function $\psi \in E$ that is* optimal *in the sense that for all $\phi \in E$ there is a constant c_ϕ with $\phi \leq 2c_\phi \psi$.*

Proof. Apply the operation $e \mapsto e'$ of theorem 4.1.1 to the sequence $\phi_e(x, y, \mu)$ ($e = 1, 2, \ldots$) of all lower semicomputable functions of x, y, μ. The elements of the sequence $\phi'_e(x, y, \mu)$, $e = 1, 2, \ldots$ are in E and and the sequence $2\phi'_e(x, y, \mu)$, $e = 1, 2, \ldots$ contains all elements of E. Hence the function $\psi(x, y, \mu) = \sum_{e=1}^{\infty} 2^{-e} \phi'_e(x, y, \mu)$ is in E and has the optimality property. □

Definition 4.1.4 A uniform test u is called *universal* if for every other test t there is a constant $c_t > 0$ such that for all x, μ we have $t_\mu(x) \leq c u_\mu(x)$. ⌟

Theorem 4.1.2 (Universal test, by Hoyrup-Rojas) *There is a universal uniform test.*

Proof. This is a special case of Corollary 4.1.3 with $s(y, \mu) = 1$. □

Definition 4.1.5 Let us fix a universal uniform test, called $\mathbf{t}_\mu(x)$. An element $x \in X$ is called *random* with respect to measure $\mu \in \mathcal{M}(\mathbf{X})$ if $\mathbf{t}_\mu(x) < \infty$.

The *deficiency of randomness* is defined as $\mathbf{d}_\mu(x) = \log \mathbf{t}_\mu(x)$. ⌟

If the space is discrete then typically all elements are random with respect to μ, but they will still be distinguished according to their different values of $\mathbf{d}_\mu(x)$.

4.1.3 Sequences

Let our computable metric space $\mathbf{X} = (X, d, D, \alpha)$ be the Cantor space of Example B.1.35.2: the set of sequences over a (finite or countable) alphabet $\Sigma^{\mathbb{N}}$. We may want to measure the non-randomness of finite sequences, viewing them as initial segments of infinite sequences. Take the universal test $\mathbf{t}_\mu(x)$. For this, it is helpful to apply the representation of Proposition B.1.23, taking into account that adding the extra parameter μ does not change the validity of the theorem:

Proposition 4.1.6 *There is a function $g : \mathcal{M}(\mathbf{X})\times\Sigma^* \to \overline{\mathbb{R}}_+$ with $\mathbf{t}_\mu(\xi) = \sup_n g_\mu(\xi^{\leq n})$, and with the following properties:*

a) *g is lower semicomputable.*

b) *$v \sqsubseteq w$ implies $g_\mu(v) \leq g_\mu(w)$.*

c) *For all integer $n \geq 0$ we have $\sum_{w\in\Sigma^n} \mu(w)g_\mu(w) \leq 1$.*

The properties of the function $g_\mu(w)$ clearly imply that $\sup_n g_\mu(\xi^{\leq n})$ is a uniform test.

The existence of a universal function among the functions g can be proved by the usual methods:

Proposition 4.1.7 *Among the functions $g_\mu(w)$ satisfying the properties listed in Proposition 4.1.6, there is one that dominates to within a multiplicative constant.*

These facts motivate the following definition.

Definition 4.1.8 (Extended test) Over the Cantor space, we extend the definition of a universal test $\mathbf{t}_\mu(x)$ to finite sequences as follows. We fix a function $\mathbf{t}_\mu(w)$ with $w \in \Sigma^*$ whose existence is assured by Proposition 4.1.7. For infinite sequences ξ we define $\mathbf{t}_\mu(\xi) = \sup_n \mathbf{t}_\mu(\xi^{\leq n})$. The test with values defined also on finite sequences will be called an *extended test*. ⌟

We could try to define extended tests also over arbitrary constructive metric spaces, extending them to the canonical balls, with the monotonicity property that $\mathbf{t}_\mu(v) \leq \mathbf{t}_\mu(w)$ if ball w is manifestly included in ball v. But there is nothing simple and obvious corresponding to the integral requirement (c).

Over the space $\Sigma^{\mathbb{N}}$ for a finite alphabet Σ, an extended test could also be extracted directly from a test, using the following formula (as observed by Vyugin and Shen).

Definition 4.1.9

$$\overline{u}(z) = \inf\{u(\omega) : \omega \text{ is an infinite extension of } z\}.$$

⌟

This function is lower semicomputable, by Proposition B.1.31.

4.1.4 Conservation of randomness

For $i = 1, 0$, let $\mathbf{X}_i = (X_i, d_i, D_i, \alpha_i)$ be computable metric spaces, and let $\mathbf{M}_i = (\mathcal{M}(\mathbf{X}_i), \sigma_i, \nu_i)$ be the effective topological space of probability measures over $\mathbf{X}_i$. Let Λ be a computable probability kernel from $\mathbf{X}_1$ to $\mathbf{X}_0$ as defined in Subsection B.2.3. In the following theorem, the same notation $\mathbf{d}_\mu(x)$ will refer to the deficiency of randomness with respect to two different spaces, $\mathbf{X}_1$ and $\mathbf{X}_0$, but this should not cause confusion. Let us first spell out the conservation theorem before interpreting it.

Theorem 4.1.3 *For a computable probability kernel Λ from $\mathbf{X}_1$ to $\mathbf{X}_0$, we have*

$$\lambda^y_x \mathbf{t}_{\Lambda^*\mu}(y) \stackrel{*}{<} \mathbf{t}_\mu(x). \tag{4.1.1}$$

Proof. Let $\mathbf{t}_\nu(x)$ be the universal test over $\mathbf{X}_0$. The left-hand side of (4.1.1) can be written as

$$u_\mu = \Lambda \mathbf{t}_{\Lambda^*\mu}.$$

According to (A.2.4), we have $\mu u_\mu = (\Lambda^*\mu)\mathbf{t}_{\Lambda^*\mu}$ which is ≤ 1 since $\mathbf{t}$ is a test. If we show that $(\mu, x) \mapsto u_\mu(x)$ is lower semicomputable then the universality of $\mathbf{t}_\mu$ will imply that $u_\mu \stackrel{*}{<} \mathbf{t}_\mu$.

According to Proposition B.1.38, as a lower semicomputable function, $\mathbf{t}_\nu(y)$ can be written as $\sup_n g_n(\nu, y)$, where $(g_n(\nu, y))$ is a computable sequence of computable functions. We pointed out in Subsection B.2.3 that the function $\mu \mapsto \Lambda^*\mu$ is computable. Therefore the function $(n, \mu, x) \mapsto g_n(\Lambda^*\mu, f(x))$ is also a computable. So, $u_\mu(x)$ is the supremum of a computable sequence of computable functions and as such, lower semicomputable. □

It is easier to interpret the theorem first in the special case when $\Lambda = \Lambda_h$ for a computable function $h : X_1 \to X_0$, as in Example B.2.12. Then the theorem simplifies to the following.

Corollary 4.1.10 *For a computable function $h : X_1 \to X_0$, we have $\mathbf{d}_{h^*\mu}(h(x)) \stackrel{+}{<} \mathbf{d}_\mu(x)$.*

Informally, this says that if x is random with respect to μ in $\mathbf{X}_1$ then $h(x)$ is essentially at least as random with respect to the output distribution $h^*\mu$ in $\mathbf{X}_0$. Decrease in randomness can only be caused by complexity in the definition of the function h.

Let us specialize the theorem even more:

Corollary 4.1.11 *For a probability distribution μ over the space $X \times Y$ let μ_X be its marginal on the space X. Then we have*

$$\mathbf{d}_{\mu_X}(x) \stackrel{+}{<} \mathbf{d}_\mu((x, y)).$$

This says, informally, that if a pair is random then each of its elements is random (with respect to the corresponding marginal distribution).

In the general case of the theorem, concerning random transitions, we cannot bound the randomness of each outcome uniformly. The theorem asserts that the average nonrandomness, as measured by the universal test with respect to the output distribution, does not increase. In logarithmic notation: $\lambda_x^y 2^{\mathbf{d}_{\Lambda^*\mu}(y)} \stackrel{+}{<} \mathbf{d}_\mu(x)$, or equivalently, $\int 2^{\mathbf{d}_{\Lambda^*\mu}(y)} \lambda_x(dy) \stackrel{+}{<} \mathbf{d}_\mu(x)$.

Corollary 4.1.12 *Let Λ be a computable probability kernel from $\mathbf{X}_1$ to $\mathbf{X}_0$. There is a constant c such that for every $x \in \mathbf{X}^1$, and integer $m > 0$ we have*

$$\lambda_x\{y : \mathbf{d}_{\Lambda^*\mu}(y) > \mathbf{d}_\mu(x) + m + c\} \leq 2^{-m}.$$

Thus, in a computable random transition, the probability of an increase of randomness deficiency by m units (plus a constant c) is less than 2^{-m}. The constant c comes from the description complexity of the transition Λ.

A randomness conservation result related to Corollary 4.1.10 was proved in [25]. There, the measure over the space $\mathbf{X}_0$ is not the output measure of the transformation, but is assumed to obey certain inequalities related to the transformation.

4.2 Test for a class of measures

4.2.1 From a uniform test

A Bernoulli measure is what we get by tossing a (possibly biased) coin repeatedly.

Definition 4.2.1 Let $X = \mathbb{B}^{\mathbb{N}}$ be the set of infinite binary sequences, with the usual sequence topology. Let B_p be the measure on X that corresponds to tossing a coin independently with probability p of success: it is called the *Bernoulli* measure with parameter p. Let $\mathcal{B}$ denote the set of all Bernoulli measures. ⌟

Given a sequence $x \in X$ we may ask the question whether x is random with respect to at least *some* Bernoulli measure. (It can clearly not be random with respect to two different Bernoulli measures since if x is random with respect to B_p then its relative frequencies converge to p.) This idea suggests two possible definitions for a test of the property of "Bernoulliness":

1. We could define $\mathbf{t}_{\mathcal{B}}(x) = \inf_{\mu\in\mathcal{B}}(x)$.
2. We could define the notion of a Bernoulli test as a lower semicomputable function $f(x)$ with the property $B_p f \leq 1$ for all p.

We will see that in case of this class of measures the two definitions lead to essentially the same test.

Let us first extend the definition to more general sets of measures, still having a convenient property.

Definition 4.2.2 (Class tests) Consider a class $\mathcal{C}$ of measures that is effectively compact in the sense of Definition B.1.27 or (equivalently for metric spaces) in the sense of Theorem B.1.1. A lower semicomputable function $f(x)$ is called a $\mathcal{C}$-*test* if for all $\mu \in \mathcal{C}$ we have $\mu f \leq 1$. It is a *universal* $\mathcal{C}$-*test* if it dominates all other $\mathcal{C}$-tests to within a multiplicative constant. ⌟

Example 4.2.3 It is easy to show that the class $\mathcal{B}$ is effectively compact. One way is to appeal to the general theorem in Proposition B.1.32 saying that applying a computable function to an effectively compact set (in this case the interval $[0, 1]$), the image is also an effectively compact set. ⌟

For the case of infinite sequences, we can also define extended tests.

Definition 4.2.4 (Extended class test) Let our space $\mathbf{X}$ be the Cantor space of infinite sequences $\Sigma^{\mathbb{N}}$. Consider a class $\mathcal{C}$ of measures that is effectively compact in the sense of Definition B.1.27 or (equivalently for metric spaces) in the sense of Theorem B.1.1. A lower semicomputable function $f : \Sigma^* \to \overline{\mathbb{R}}_+$ is called an *extended* $\mathcal{C}$-*test* if it is monotonic with respect to the prefix relation and for all $\mu \in \mathcal{C}$ and integer $n \geq 0$ we have

$$\sum_{x \in \Sigma^n} \mu(x) f(x) \leq 1.$$

It is *universal* if it dominates all other extended $\mathcal{C}$-tests to within a multiplicative constant. ⌟

The following observation is immediate.

Proposition 4.2.5 *A function $f : \Sigma^{\mathbb{N}} \to \overline{\mathbb{R}}_+$ is a class test if and only if it can be represented as $\lim_n g(\xi^{\leq n})$ where $g(x)$ is an extended class test.*

The following theorem defines a universal $\mathcal{C}$-test.

Theorem 4.2.1 *Let $\mathbf{t}_\mu(x)$ be a universal uniform test. Then $u(x) = \inf_{\mu \in \mathcal{C}} \mathbf{t}_\mu(x)$ defines a universal $\mathcal{C}$-test.*

Proof. Let us show first that $u(x)$ is a $\mathcal{C}$-test. It is lower semicomputable according to Proposition B.1.31. Also, for each μ we have $\mu u \leq \mu \mathbf{t}_\mu \leq 1$, showing that $u(x)$ is a $\mathcal{C}$-test.

Let us now show that it is universal. Let $f(x)$ be an arbitrary $\mathcal{C}$-test. By Corollary 4.1.2 there is a uniform test $g_\mu(x)$ such that for all $\mu \in \mathcal{C}$ we have $g_\mu(x) = f(x)/2$. It follows from the universality of the uniform test $\mathbf{t}_\mu(x)$ that $f(x) \stackrel{*}{<} g_\mu(x) \stackrel{*}{<} \mathbf{t}_\mu(x)$ for all $\mu \in \mathcal{C}$. But then $f(x) \stackrel{*}{<} \inf_{\mu \in \mathcal{C}} \mathbf{t}_\mu(x) = u(x)$. □

For the case of sequences, the same statement can be made for extended tests. (This is not completely automatic since a test is obtained from an extended test via a supremum, on the other hand a class test is obtained, according the theorem above, via an infimum.)

4.2.2 Typicality and class tests

The set of Bernoulli measures has an important property shared by many classes considered in practice: namely that random sequences determine the measure to which it belongs.

Definition 4.2.6 Consider a class $\mathcal{C}$ of measures over a computable metric space $\mathbf{X} = (X, d, D, \alpha)$. We will say that a lower semicomputable function

$$s : X \times \mathcal{C} \to \overline{\mathbb{R}}_+$$

is a *separating test* for $\mathcal{C}$ if

- $s_\mu(\cdot)$ is a test for each $\mu \in \mathcal{C}$.
- If $\mu \neq \nu$ then $s_\mu(x) \vee s_\nu(x) = \infty$ for all $x \in X$.

Given a separating test $s_\mu(x)$ we call an element x *typical* for $\mu \in \mathcal{C}$ if $s_\mu(x) < \infty$. ⌟

A typical element determines uniquely the measure μ for which it is typical. Note that if a separating tests exists for a class then any two different measures μ_1, μ_2 in the class are orthogonal to each other, that is there are disjoint measureable sets A_1, A_2 with $\mu_j(A_j) = 1$. Indeed, let $A_j = \{x : s_{\mu_j}(x) < \infty\}$.

Let us show a nontrivial example: the class of $\mathcal{B}$ of Bernoulli measures. Recall that by Chebyshev's inequality we have

$$B_p(\{x \in \mathbb{B}^n : |\sum_i x(i) - np| \geq \lambda n^{1/2}(p(1-p))^{1/2}\}) \leq \lambda^{-2}.$$

Since $p(1-p) \leq 1/4$, this implies

$$B_p(\{x \in \mathbb{B}^n : |\sum_i x(i) - np > \lambda n^{1/2}/2\}) < \lambda^{-2}.$$

Setting $\lambda = n^{0.1}$ and ignoring the factor $1/2$ gives

$$B_p(\{x \in \mathbb{B}^n : |\sum_i x(i) - np| > n^{0.6}\}) < n^{-0.2}.$$

Setting $n = 2^k$:

$$B_p(\{x \in \mathbb{B}^{2^k} : |\sum_i x(i) - 2^k p| > 2^{0.6k}\}) < 2^{-0.2k}. \tag{4.2.1}$$

Now, for the example.

Example 4.2.7 For a sequence ξ in $\mathbf{B}^{\mathbb{N}}$, and for $p \in [0, 1]$ let

$$g_p(x) = g_{B_p}(x) = \sup\{k : |\sum_{i=1}^{2^k} \xi(i) - 2^k p| > 2^{0.6k}\}.$$

Then we have

$$B_p^\xi g_p(\xi) \leq \sum_k k \cdot 2^{-0.2k} = c < \infty$$

for some constant c, so $s_p(\xi) = g_p(x)/c$ is a test for each p. The property $s_p(\xi) < \infty$ implies that $2^{-k} \sum_{i=1}^{2^k} \xi(i)$ converges to p. For a given ξ this is impossible for both p and q for $p \neq q$, hence $s_p(\xi) \vee s_q(\xi) = \infty$. ⌟

The following structure theorem gives considerable insight.

Theorem 4.2.2 *Let $\mathcal{C}$ be an effectively compact class of measures, let $\mathbf{t}_\mu(x)$ be the universal uniform test and let $\mathbf{t}_\mathcal{C}(x)$ be a universal class test for $\mathcal{C}$. Assume that a separating test $s_\mu(x)$ exists for $\mathcal{C}$. Then we have the representation*

$$\mathbf{t}_\mu(x) \stackrel{*}{=} \mathbf{t}_\mathcal{C}(x) \vee s_\mu(x)$$

for all $\mu \in \mathcal{C}$, $x \in X$.

Proof. First, we have $\mathbf{t}_\mathcal{C}(x) \vee s_\mu(x) \stackrel{*}{<} \mathbf{t}_\mu(x)$. Indeed as we know from the Uniform Extension Corollary 4.1.2, we can extend $s_\mu(x)/2$ to a uniform test, hence $s_\mu(x) \stackrel{*}{<} \mathbf{t}_\mu(x)$. Also by definition $\mathbf{t}_\mathcal{C}(x) \leq \mathbf{t}_\mu(x)$.

On the other hand, let us show $\mathbf{t}_\mathcal{C}(x) \vee s_\mu(x) \geq \mathbf{t}_\mu(x)$. Suppose first that x is not random with respect to any $\nu \in \mathcal{C}$: then $\mathbf{t}_\mathcal{C}(x) = \infty$. Suppose now that x is random with respect to some $\nu \in \mathcal{C}$, $\nu \neq \mu$. Then $s_\mu(x) = \infty$. Finally, suppose $\mathbf{t}_\mu(x) < \infty$. Then $\mathbf{t}_\nu(x) = \infty$ for all $\nu \in \mathcal{C}$, $\nu \neq \mu$, hence $\mathbf{t}_\mathcal{C}(x) = \inf_{\nu \in \mathcal{C}} \mathbf{t}_\nu(x) = \mathbf{t}_\mu(x)$, so the inequality holds again. □

The above theorem separates the randomness test into two parts. One part tests randomness with respect to the class $\mathcal{C}$, the other one tests typicality with respect to the measure μ. In the Bernoulli example,

- Part $\mathbf{t}_\mathcal{B}(\xi)$ checks "Bernoulliness", that is independence. It encompasses all the irregularity criteria.
- Part $s_p(\xi)$ checks (crudely) for the law of large numbers: whether relative frequency converges (fast) to p.

If the independence of the sequence is taken for granted, we may assume that the class test is satisfied. What remains is typicality testing, which is similar to ordinary statistical parameter testing.

Remarks 4.2.8 1. Separation is the only requirement of the test $s_\mu(x)$, otherwise, for example in the Bernoulli test case, no matter how crude the convergence criterion expressed by $s_\mu(x)$, the maximum $\mathbf{t}_{\mathcal{C}}(x) \vee s_\mu(x)$ is always (essentially) the same universal test.

2. Though the convergence criterion can be crude, but one still seems to need some kind of constructive convergence of the relative frequencies if the separation test is to be defined in terms of relative frequency convergence.

⌟

Example 4.2.9 For $\varepsilon > 0$ let $P(\varepsilon)$ be the Markov chain $X_1, X_2, \ldots$ with set of states $\{0, 1\}$, with transition probabilities $T(0, 1) = T(1, 0) = \varepsilon$ and $T(0, 0) = T(1, 1) = 1 - \varepsilon$, and with $P[X_1 = 0] = P[X_1 = 1] = 1/2$. Let $\mathcal{C}(\delta)$ be the class of all $P(\varepsilon)$ with $\varepsilon \geq \delta$. For each $\delta > 0$, this is an effectively compact class, and a separating test is easy to construct since an effective law of large numbers holds for these Markov chains.

We can generalize to the set of m-state stationary Markov chains whose eigenvalue gap is $\geq \varepsilon$. ⌟

This example is in contrast to V'yugin's example [54] showing that, in the nonergodic case, in general no recursive speed of convergence can be guaranteed in the Ergodic Theorem (which is the appropriate generalization of the law of large numbers)[1].

We can show that if a computable Markov chain is ergodic then its law of large numbers *does* have a constructive speed of convergence. Hopefully, this observation can be extended to some interesting compact classes of ergodic processes.

4.2.3 Martin-Löf's approach

Martin-Löf also gave a definition of Bernoulli tests in [37]. For its definition let us introduce the following notation.

Notation 4.2.10 The set of sequences with a given frequency of 1's will be denoted as follows:

$$\mathbb{B}(n, k) = \{x \in \mathbb{B}^n : \sum_i x(i) = k\}.$$

⌟

Martin-Löf's definition translates to the integral constraint version as follows:

[1] This paper of Vyugin also shows that though there is no recursive speed of convergence, a certain constructive proof of the pointwise ergodic theorem still gives rise to a test of randomness.

Definition 4.2.11 Let $X = \mathbb{B}^{\mathbb{N}}$ be the set of infinite binary sequences with the usual metrics. A *combinatorial Bernoulli test* is a function $f : \mathbb{B}^* \to \overline{\mathbb{R}}_+$ with the following constraints:

a) It is lower semicomputable.

b) It is monotonic with respect to the prefix relation.

c) For all $0 \leq k \leq n$ we have

$$\sum_{x \in \mathbb{B}(n,k)} f(x) \leq \binom{n}{k}. \tag{4.2.2}$$

⌟

The following observation is useful.

Proposition 4.2.12 *If a combinatorial Bernoulli test $f(x)$ is given on strings x of length less than n, then extending it to longer strings using monotonicity we get a function that is still a combinatorial Bernoulli test.*

Proof. It is sufficient to check the relation (4.2.2). We have (even for $k = 0$, when $\mathbb{B}(n-1, k-1) = \emptyset$):

$$\begin{aligned}\sum_{x \in \mathbb{B}(n,k)} f(x) &\leq \sum_{y \in \mathbb{B}(n-1,k-1)} f(y) + \sum_{y \in \mathbb{B}(n-1,k)} f(y) \\ &\leq \binom{n-1}{k-1} + \binom{n-1}{k} = \binom{n}{k}.\end{aligned}$$

□

The following can be shown using standard methods:

Proposition 4.2.13 (Universal combinatorial Bernoulli test) *There is a universal combinatorial Bernoulli test $f(x)$, that is a combinatorial Bernoulli test with the property that for every combinatorial Bernoulli test $h(x)$ there is a constant $c_h > 0$ such that for all x we have $h(x) \leq c_h g(x)$.*

Definition 4.2.14 Let us fix a universal combinatorial Bernoulli test $b(x)$ and extend it to infinite sequences ξ by

$$b(\xi) = \sup_n b(\xi^{\leq n}).$$

Let $\mathbf{t}_{\mathcal{B}}(\xi)$ be a universal class test for Bernoulli measures, for $\xi \in \mathbb{B}^{\mathbb{N}}$. ⌟

Let us show that the general class test for Bernoulli measures and Martin-Löf's Bernoulli test yield the same random sequences.

Theorem 4.2.3 *With the above definitions, we have $b(\xi) \stackrel{*}{=} \mathbf{t}_{\mathcal{B}}(\xi)$. In words: a sequence is nonrandom with respect to all Bernoulli measures if and only if it is rejected by a universal combinatorial Bernoulli test.*

Proof. We first show $b(\xi) \leq \mathbf{t}_{\mathcal{B}}(\xi)$. Moreover, we show that $b(x)$ is an extended class test for the class of Bernoulli measures. We only need to check the sum condition, namely that for all $0 \leq p \leq 1$, and all $n > 0$ the inequality $\sum_{x \in \mathbb{B}^n} B_p(x) b(x) \leq 1$ holds. Indeed, we have

$$\begin{aligned} \sum_{x \in \mathbb{B}^n} B_p(x) f(x) &= \sum_{k=0}^{n} p^k (1-p)^{n-k} \sum_{x \in \mathbb{B}(n,k)} f(x) \\ &\leq \sum_{k=0}^{n} p^k (1-p)^{n-k} \binom{n}{k} = 1. \end{aligned}$$

On the other hand, let $f(x) = \mathbf{t}_{\mathcal{B}}(x)$, $x \in \mathbb{B}^*$ be the extended test for $\mathbf{t}_{\mathcal{B}}(\xi)$. For all integers $N > 0$ let $n = \lfloor \sqrt{N/2} \rfloor$. Then as N runs through the integers, n also runs through all integers. For $x \in \mathbb{B}^N$ let $F(x) = f(x^{\leq n})$. Since $f(x)$ is lower semicomputable and monotonic with respect to the prefix relation, this is also true of $F(x)$.

We need to estimate $\sum_{x \in \mathbb{B}(N,K)} F(x)$. For this, note that for $y \in \mathbb{B}(n,k)$ we have

$$|\{x \in \mathbb{B}(N,K) : y \sqsubseteq x\}| = \binom{N-n}{K-k}. \tag{4.2.3}$$

Now for $0 \leq K \leq N$ we have

$$\begin{aligned} \sum_{x \in \mathbb{B}(N,K)} F(x) &= \sum_{y \in \mathbb{B}^n} f(y) |\{x \in \mathbb{B}(N,K) : y \sqsubseteq x\}| \\ &= \sum_{k=0}^{n} \binom{N-n}{K-k} \sum_{y \in \mathbb{B}(n,k)} f(y). \end{aligned} \tag{4.2.4}$$

Let us estimate $\binom{N-n}{K-k} / \binom{N}{K}$. If $K = k = 0$ then this is 1. If $k = n$ then it is $\frac{K \cdots (K-n+1)}{N \cdots (N-n+1)}$. Otherwise, using $p = K/N$:

$$\begin{aligned} \frac{\binom{N-n}{K-k}}{\binom{N}{K}} &= \frac{(N-n)(N-n-1)\cdots(N-K-(n-k)+1)/(K-k)!}{N(N-1)\cdots(N-K+1)/K!} \\ &= \frac{K \cdots (K-k+1) \cdot (N-K) \cdots (N-K-(n-k)+1)}{N \cdots (N-n+1)}. \\ &\leq \frac{K^k (N-K)^{n-k}}{(N-n)^n} = p^k (1-p)^{n-k} \left(\frac{N}{N-n} \right)^n. \end{aligned} \tag{4.2.5}$$

Thus in all cases the estimate

$$\binom{N-n}{K-n}\Big/\binom{N}{K} \le p^k(1-p)^{n-k}\left(\frac{N}{N-n}\right)^n$$

holds. We have

$$\left(\frac{N}{N-n}\right)^n = \left(1+\frac{n}{N-n}\right)^n \le e^{\frac{N}{2(N-n)}} \le e^2,$$

since we assumed $2n^2 \le N$. Substituting into (4.2.4) gives

$$\sum_{x\in\mathbb{B}(N,K)} F(x) \le e^2 \sum_{k=0}^{n} p^k(1-p)^{n-k} \sum_{y\in\mathbb{B}(n,k)} f(y) \le e^2,$$

since $f(x)$ is an extended class test for Bernoulli measures. It follows that $e^{-2}F(x) \stackrel{*}{<} b(x)$, hence also $F(x) \stackrel{*}{<} b(x)$. But we have $\mathbf{t}_{\mathcal{C}}(\xi) = \sup_n f(\xi^{\le n}) = \sup_n F(\xi^{\le n})$, hence $\mathbf{t}_{\mathcal{C}}(\xi) \stackrel{*}{<} b(\xi)$. □

4.3 Neutral measure

Let $\mathbf{t}_\mu(x)$ be our universal uniform randomness test. We call a measure M *neutral* if $\mathbf{t}_M(x) \le 1$ for all x. If M is neutral then no experimental outcome x could refute the theory (hypothesis, model) that M is the underlying measure to our experiments. It can be used as "apriori probability", in a Bayesian approach to statistics. Levin's theorem says the following:

Theorem 4.3.1 *If the space* $\mathbf{X}$ *is compact then there is a neutral measure over* $\mathbf{X}$.

The proof relies on a nontrivial combinatorial fact, Sperner's Lemma, which also underlies the proof of the Brouwer fixpoint theorem. Here is a version of Sperner's Lemma, spelled out in continuous form:

Proposition 4.3.1 *Let* $p_1, \ldots, p_k$ *be points of some finite-dimensional space* $\mathbb{R}^n$. *Suppose that there are closed sets* $F_1, \ldots, F_k$ *with the property that for every subset* $1 \le i_1 < \cdots < i_j \le k$ *of the indices, the simplex* $S(p_{i_1}, \ldots, p_{i_j})$ *spanned by* $p_{i_1}, \ldots, p_{i_j}$ *is covered by the union* $F_{i_1} \cup \cdots \cup F_{i_j}$. *Then the intersection* $\bigcap_i F_i$ *of all these sets is not empty.*

The following lemma will also be needed.

Lemma 4.3.2 *For every closed set* $A \subset \mathbf{X}$ *and measure* μ, *if* $\mu(A) = 1$ *then there is a point* $x \in A$ *with* $\mathbf{t}_\mu(x) \le 1$.

Proof. This follows easily from $\mu\, t_\mu = \mu^x 1_A(x) t_\mu(x) \le 1$. □

Proof of Theorem 4.3.1. For every point $x \in \mathbf{X}$, let F_x be the set of measures for which $\mathbf{t}_\mu(x) \le 1$. If we show that for every finite set of points $x_1, \dots, x_k$, we have

$$F_{x_1} \cap \dots \cap F_{x_k} \neq \emptyset, \tag{4.3.1}$$

then we will be done. Indeed, according to Proposition A.2.49, the compactness of $\mathbf{X}$ implies the compactness of the space $\mathbf{M}(\mathbf{X})$ of measures. Therefore if every finite subset of the family $\{F_x : x \in \mathbf{X}\}$ of closed sets has a nonempty intersection, then the whole family has a nonempty intersection: this intersection consists of the neutral measures.

To show (4.3.1), let $S(x_1, \dots, x_k)$ be the set of probability measures concentrated on $x_1, \dots, x_k$. Lemma 4.3.2 implies that each such measure belongs to one of the sets F_{x_i}. Hence $S(x_1, \dots, x_k) \subset F_{x_1} \cup \dots \cup F_{x_k}$, and the same holds for every subset of the indices $\{1, \dots, k\}$. Sperner's Lemma 4.3.1 implies $F_{x_1} \cap \dots \cap F_{x_k} \neq \emptyset$. □

When the space is not compact, there are generally no neutral probability measures, as shown by the following example.

Proposition 4.3.3 *Over the discrete space $\mathbf{X} = \mathbb{N}$ of natural numbers, there is no neutral measure.*

Proof. It is sufficient to construct a randomness test $t_\mu(x)$ with the property that for every measure μ, we have $\sup_x t_\mu(x) = \infty$. Let

$$t_\mu(x) = \sup\{k \in \mathbb{N} : \sum_{y<x} \mu(y) > 1 - 2^{-k}\}. \tag{4.3.2}$$

By its construction, this is a lower semicomputable function with $\sup_x t_\mu(x) = \infty$. It is a test if $\sum_x \mu(x) t_\mu(x) \le 1$. We have

$$\sum_x \mu(x) t_\mu(x) = \sum_{k>0} \sum_{t_\mu(x) \ge k} \mu(x) < \sum_{k>0} 2^{-k} \le 1.$$

□

Using a similar construction over the space $\mathbb{N}^\mathbb{N}$ of infinite sequences of natural numbers, we could show that for every measure μ there is a sequence x with $\mathbf{t}_\mu(x) = \infty$.

Proposition 4.3.3 is a little misleading, since $\mathbb{N}$ can be compactified into $\overline{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$ (as in Part 1 of Example A.1.22). Theorem 4.3.1 implies that there is a neutral probability measure M over the compactified space $\overline{\mathbb{N}}$. Its restriction to $\mathbb{N}$ is, of course, not a probability measure, since it satisfies only $\sum_{x<\infty} M(x) \le 1$. We called these functions *semimeasures*.

Remark 4.3.4

1. It is easy to see that Theorem 4.6.1 characterizing randomness in terms of complexity holds also for the space $\overline{\mathbb{N}}$.
2. The topological space of semimeasures over $\mathbb{N}$ is not compact, and there is no neutral one among them. Its topology is not the same as what we get when we restrict the topology of probability measures over $\overline{\mathbb{N}}$ to $\mathbb{N}$. The difference is that over $\mathbb{N}$, for example the set of measures $\{\mu : \underline{\mu}(\mathbb{N}) > 1/2\}$ is closed, since $\mathbb{N}$ (as the whole space) is a closed set. But over $\overline{\mathbb{N}}$, this set is not necessarily closed, since $\mathbb{N}$ is not a closed subset of $\overline{\mathbb{N}}$.

⌟

Neutral measures are not too simple, even over $\overline{\mathbb{N}}$, as the following theorem shows.

Theorem 4.3.2 *There is no neutral measure over $\overline{\mathbb{N}}$ that is upper semicomputable over $\mathbb{N}$ or lower semicomputable over $\mathbb{N}$.*

Proof. Let us assume that ν is a measure that is upper semicomputable over $\mathbb{N}$. Then the set

$$\{(x, r) : x \in \mathbb{N},\ r \in \mathbb{Q},\ \nu(x) < r\}$$

is recursively enumerable: let (x_i, r_i) be a particular enumeration. For each n, let $i(n)$ be the first i with $r_i < 2^{-n}$, and let $y_n = x_{i(n)}$. Then $\nu(y_n) < 2^{-n}$, and at the same time $H(y_n) \stackrel{+}{<} H(n)$. As mentioned, in Remark 4.3.4, Theorem 4.6.1 characterizing randomness in terms of complexity holds also for the space $\overline{\mathbb{N}}$. Thus,

$$\mathbf{d}_\nu(y_n) \stackrel{+}{=} -\log \nu(y_n) - H(y_n \mid \nu) \stackrel{+}{>} n - H(n).$$

Suppose now that ν is lower semicomputable over $\mathbb{N}$. The proof for this case is longer. We know that ν is the monotonic limit of a recursive sequence $i \mapsto \nu_i(x)$ of recursive semimeasures with rational values $\nu_i(x)$. For every $k = 0, \ldots, 2^n - 2$, let

$$V_{n,k} = \{\mu \in \mathcal{M}(\overline{\mathbb{N}}) : k \cdot 2^{-n} < \mu(\{0, \ldots, 2^n - 1\}) < (k+2) \cdot 2^{-n}\},$$
$$J = \{(n, k) : k \cdot 2^{-n} < \nu(\{0, \ldots, 2^n - 1\})\}.$$

The set J is recursively enumerable. Let us define the functions $j : J \to \mathbb{N}$ and $x : J \to \{0, \ldots, 2^n - 1\}$ as follows: $j(n, k)$ is the smallest i with $\nu_i(\{0, \ldots, 2^n - 1\}) > k \cdot 2^{-n}$, and

$$x_{n,k} = \min\{y < 2^n : \nu_{j(n,k)}(y) < 2^{-n+1}\}.$$

Let us define the function $f_\mu(x, n, k)$ as follows. We set $f_\mu(x, n, k) = 2^{n-2}$ if the following conditions hold:

a) $\mu \in V_{n,k}$;

b) $\mu(x) < 2^{-n+2}$;

c) $(n,k) \in J$ and $x = x_{n,k}$.

Otherwise, $f_\mu(x,n,k) = 0$. Clearly, the function $(\mu, x, n, k) \mapsto f_\mu(x,n,k)$ is lower semicomputable. Condition (b) implies

$$\sum_y \mu(y) f_\mu(y,n,k) \le \mu(x_{n,k}) f_\mu(x_{n,k},n,k) < 2^{-n+2} \cdot 2^{n-2} = 1. \tag{4.3.3}$$

Let us show that $\nu \in V_{n,k}$ implies

$$f_\nu(x_{n,k},n,k) = 2^{n-2}. \tag{4.3.4}$$

Consider $x = x_{n,k}$. Conditions (a) and (c) are satisfied by definition. Let us show that condition (b) is also satisfied. Let $j = j(n,k)$. By definition, we have $\nu_j(x) < 2^{-n+1}$. Since by definition $\nu_j \in V_{n,k}$ and $\nu_j \le \nu \in V_{n,k}$, we have

$$\nu(x) \le \nu_j(x) + 2^{-n+1} < 2^{-n+1} + 2^{-n+1} = 2^{-n+2}.$$

Since all three conditions (a), (b) and (c) are satisfied, we have shown (4.3.4). Now we define

$$g_\mu(x) = \sum_{n \ge 2} \frac{1}{n(n+1)} \sum_k f_\mu(x,n,k).$$

Let us prove that $g_\mu(x)$ is a uniform test. It is lower semicomputable by definition, so we only need to prove $\sum_x \mu(x) f_\mu(x) \le 1$. For this, let $I_{n,\mu} = \{k : \mu \in V_{n,k}\}$. Clearly by definition, $|I_{n,\mu}| \le 2$. We have, using this last fact and the test property (4.3.3):

$$\sum_x \mu(x) g_\mu(x) = \sum_{n \ge 2} \frac{1}{n(n+1)} \sum_{k \in I_{n,\mu}} \sum_x \mu(x) f_\mu(x,n,k) \le \sum_{n \ge 2} \frac{1}{n(n+1)} \cdot 2 \le 1.$$

Thus, $g_\mu(x)$ is a uniform test. If $\nu \in V_{n,k}$ then we have

$$\mathbf{t}_\nu(x_{n,k}) \stackrel{*}{>} g_\nu(x_{n,k}) \ge \frac{1}{n(n+1)} f_\mu(x_{n,k},n,k) \ge \frac{2^{n-2}}{n(n+1)}.$$

Hence ν is not neutral. □

Remark 4.3.5 In [31] and [32], Levin imposed extra conditions on tests which allow to find a lower semicomputable neutral semimeasure. ⌟

The universal lower semicomputable semimeasure $\mathbf{m}(x)$ has a certain property similar to neutrality. According to Theorem 2.3.4 specialized to one-element sequences, for every computable measure μ we have $\mathbf{d}_\mu(x) \stackrel{+}{=} -\log \mu(x) - H(x)$ (where the constant in $\stackrel{+}{=}$ depends on μ). So, for computable measures, the expression

$$\overline{\mathbf{d}}_\mu(x) = -\log \mu(x) - H(x) \tag{4.3.5}$$

can serve as a reasonable deficiency of randomness. (We will also use the test $\overline{\mathbf{t}} = 2^{\overline{\mathbf{d}}}$.) If we substitute $\mathbf{m}$ for μ in $\overline{\mathbf{d}}_\mu(x)$, we get 0. This substitution is not justified, of course. The fact that $\mathbf{m}$ is not a probability measure can be helped, using compactification as above, and extending the notion of randomness tests. But the test $\overline{\mathbf{d}}_\mu$ can replace $\mathbf{d}_\mu$ only for computable μ, while $\mathbf{m}$ is not computable. Anyway, this is the sense in which all outcomes might be considered random with respect to $\mathbf{m}$, and the heuristic sense in which $\mathbf{m}$ may be considered "neutral".

4.4 Monotonicity, quasi-convexity/concavity

Some people find that μ-tests as defined in Definition 4.1.1 are too general, in case μ is a non-computable measure. In particular, randomness with respect to computable measures has a certain—intuitively meaningful—monotonicity property: roughly, if ν is greater than μ then if x is random with respect to μ, it should also be random with respect to ν.

Proposition 4.4.1 *For computable measures μ, ν we have for all rational $c > 0$:*

$$2^{-k}\mu \le \nu \Rightarrow \mathbf{d}_\nu(x) \stackrel{+}{<} \mathbf{d}_\mu(x) + k + H(k). \tag{4.4.1}$$

Here the constant in $\stackrel{+}{<}$ depends on μ, ν, but not on k.

Proof. We have $1 \ge \nu\mathbf{t}_\nu \ge 2^{-k}\mu\mathbf{t}_\nu$, hence $2^{-k}\mathbf{t}_\nu$ is a μ-test. Using the method of Theorem 4.1.1 in finding universal tests, one can show that the sum

$$\sum_{k:2^{-k}\mu\mathbf{t}_\nu<1} 2^{-k-H(k)}\mathbf{t}_\nu$$

is a μ-test, and hence $\stackrel{*}{<} \mathbf{t}_\mu$. Therefore this is true of each member of the sum, which is just what the theorem claims. □

There are other properties true for tests on computable measures that we may want to require for all measures. For the following properties, let us define quasi-convexity, which is a weakening of the notion of convexity.

Definition 4.4.2 A function $f : V \to \mathbb{R}$ defined on a vector space V is called *quasi-convex* if for every real number x the set $\{v : f(v) \leq x\}$ is convex. It is *quasi-concave* if $-f$ is quasi-convex. ⌟

It is easy to see that quasi-convexity is equivalent to the inequality

$$f(\lambda u + (1-\lambda)v) \leq f(u) \vee f(v)$$

for all u, v and $0 < \lambda < 1$, while quasi-concavity is equivalent to

$$f(\lambda u + (1-\lambda)v) \geq f(u) \wedge f(v)$$

The uniform test with respect to computable measures is approximately both quasi-convex and quasi-concave. Let $\nu = \lambda\mu_1 + (1-\lambda)\mu_2$.

Quasi-convexity means, roughly, that if x is random with respect to both μ_1 and μ_2 then it is also random with respect to ν. This property strengthens monotonicity in the cases where it applies.

Proposition 4.4.3 *Let μ_1, μ_2 be computable measures and $0 < \lambda < 1$ computable, with $\nu = \lambda\mu_1 + (1-\lambda)\mu_2$. Then we have*

$$\mathbf{d}_\nu(x) \stackrel{+}{<} \mathbf{d}_{\mu_1}(x) \vee \mathbf{d}_{\mu_2}(x) + H(\lambda).$$

Proof. The relation $1 \geq \nu\mathbf{t}_\nu = \lambda\mu_1\mathbf{t}_\nu + (1-\lambda)\mu_2\mathbf{t}_\nu$ implies $1 \geq \mu_i\mathbf{t}_\nu$ for some $i \in \{1, 2\}$. Then $\mathbf{d}_\nu \stackrel{+}{<} \mathbf{d}_{\mu_i} + H(\lambda)$ (since λ was used to define ν and thus $\mathbf{t}_\nu$). □

Quasi-concavity means, roughly, that if x is non-random with respect to both μ_1 and μ_2 then it is also nonrandom with respect to ν:

Proposition 4.4.4 *Let μ_1, μ_2 be computable measures and $0 < \lambda < 1$ arbitrary (not even necessarily computable), with $\nu = \lambda\mu_1 + (1-\lambda)\mu_2$. Then we have*

$$\mathbf{d}_\nu \stackrel{+}{>} \mathbf{d}_{\mu_1} \wedge \mathbf{d}_{\mu_2}.$$

Proof. The function $\mathbf{t}_{\mu_1} \wedge \mathbf{t}_{\mu_2}$ is lower semicomputable, and is a μ_i-test for each i. Therefore it is also a ν-test, and as such is $\stackrel{*}{<} \mathbf{t}_\nu$. Here, the constant in the $\stackrel{*}{<}$ depends only on (the programs for) μ_1, μ_2, and not on λ. □

These properties do not survive for arbitrary measures and arbitrary constants.

Example 4.4.5 Let measure μ_1 be uniform over the interval $[0, 1/2]$, let μ_2 be uniform over $[1/2, 1]$. For $0 \leq x \leq 1$ let

$$\nu_x = (1-x)\mu_1 + x\mu_2.$$

Then $\nu_{1/2}$ is uniform over $[0, 1]$. Let ϕ_x be the uniform distribution over $[x, x + 1/2]$, and ψ_x the uniform distribution over $[0, x] \cup [x + 1/2, 1]$.

Let $p < 1/2$ be random with respect to the uniform distribution $\nu_{1/2}$.

1. The relations

$$\nu_{1/2} \leq p^{-1}\nu_p, \quad \mathbf{d}_{\nu_{1/2}}(p) < \infty, \quad \mathbf{d}_{\nu_p}(p) = \infty$$

show that any statement analogous to the monotonicity property of Proposition 4.4.1 fails when the measures involved are not required to be computable.

2. For rational r with $0 < r < p$ the relations

$$\nu := (1 - p)\nu_r + p\nu_{1-r}, \quad \mathbf{d}_{\nu}(p) = \infty, \quad \mathbf{d}_{\nu_r}(p) < \infty, \quad \mathbf{d}_{\nu_{1-r}}(p) < \infty$$

provide a similar counterexample to Proposition 4.4.3.

3. The relations

$$\nu_{1/2} = (\nu_p + \nu_{1-p})/2, \quad \mathbf{d}_{\nu_{1/2}}(p) < \infty, \quad \mathbf{d}_{\nu_p}(p) = \mathbf{d}_{\nu_{1-p}}(p) = \infty$$

provide a similar counterexample to Proposition 4.4.4.

The following counterexample relies on a less subtle effect:

$$\nu_{1/2} = (\phi_p + \psi_p)/2, \quad \mathbf{d}_{\nu_{1/2}}(p) < \infty, \quad \mathbf{d}_{\phi_p}(p) = \mathbf{d}_{\psi_p}(p) = \infty,$$

since as a boundary point of the support, p is computable from both ϕ_p and ψ_p in a uniform way.

For a complete proof, uniform tests must be provided for each of the cases: this is left as exercise for the reader. ⌟

The non-monotonicity example could be used to argue that the we allowed too many μ-tests, that the test $\mathbf{t}_\mu(x)$ should not be allowed to depend on properties of μ that exploit the computational properties of μ so much stronger than its quantitative properties.

4.5 Algorithmic entropy

Some properties of description complexity make it a good expression of the idea of individual information content.

4.5.1 Entropy

The entropy of a discrete probability distribution μ is defined as

$$\mathcal{H}(\mu) = -\sum_x \mu(x) \log \mu(x).$$

To generalize entropy to continuous distributions the *relative entropy* is defined as follows. Let μ, ν be two measures, where μ is taken (typically, but not always), to be a probability measure, and ν another measure, that can also be a probability measure but is most frequently not. We define the *relative entropy* $\mathcal{H}_\nu(\mu)$ as follows. If μ is not absolutely continuous with respect to ν then $\mathcal{H}_\nu(\mu) = -\infty$. Otherwise, writing

$$\frac{d\mu}{d\nu} = \frac{\mu(dx)}{\nu(dx)} =: f(x)$$

for the (Radon-Nikodym) derivative (density) of μ with respect to ν, we define

$$\mathcal{H}_\nu(\mu) = -\int \log \frac{d\mu}{d\nu} d\mu = -\mu^x \log \frac{\mu(dx)}{\nu(dx)} = -\nu^x f(x) \log f(x).$$

Thus, $\mathcal{H}(\mu) = \mathcal{H}_\#(\mu)$ is a special case.

Example 4.5.1 Let $f(x)$ be a probability density function for the distribution μ over the real line, and let λ be the Lebesgue measure there. Then

$$\mathcal{H}_\lambda(\mu) = -\int f(x) \log f(x) dx.$$

⌟

In information theory and statistics, when both μ and ν are probability measures, then $-\mathcal{H}_\nu(\mu)$ is also denoted $D(\mu \parallel \nu)$, and called (after Kullback) the information divergence of the two measures. It is frequently used in the role of a distance between μ and ν. It is not symmetric, but can be shown to obey the triangle inequality, and to be nonnegative. Let us prove the latter property: in our terms, it says that relative entropy is nonpositive when both μ and ν are probability measures.

Proposition 4.5.2 *Over a space X, we have*

$$\mathcal{H}_\nu(\mu) \le -\mu(X) \log \frac{\mu(X)}{\nu(X)}. \tag{4.5.1}$$

In particular, if $\mu(X) \ge \nu(X)$ then $\mathcal{H}_\nu(\mu) \le 0$.

Proof. The inequality $-a \ln a \leq -a \ln b + b - a$ expresses the concavity of the logarithm function. Substituting $a = f(x)$ and $b = \mu(X)/\nu(X)$ and integrating by ν:

$$(\ln 2)\mathcal{H}_\nu(\mu) = -\nu^x f(x) \ln f(x) \leq -\mu(X) \ln \frac{\mu(X)}{\nu(X)} + \frac{\mu(X)}{\nu(X)}\nu(X) - \mu(X)$$
$$= -\mu(X) \ln \frac{\mu(X)}{\nu(X)},$$

giving (4.5.1). □

4.5.2 Algorithmic entropy

Le us recall some facts on description complexity. Let us fix some (finite or infinite) alphabet Σ and consider the discrete space Σ^*.

The universal lower semicomputable semimeasure $\mathbf{m}(x)$ over Σ^* was defined in Definition 1.6.9. It is possible to turn $\mathbf{m}(x)$ into a measure, by compactifying the discrete space Σ^* into

$$\overline{\Sigma^*} = \Sigma^* \cup \{\infty\}$$

(as in part 1 of Example A.1.22; this process makes sense also for a constructive discrete space), and setting $\mathbf{m}(\infty) = 1 - \sum_{x \in \Sigma^*} \mathbf{m}(x)$. The extended measure $\mathbf{m}$ is not quite lower semicomputable since the number $\mu(\overline{\Sigma^*} \setminus \{0\})$ is not necessarily lower semicomputable.

Remark 4.5.3 A measure μ is computable over $\overline{\Sigma^*}$ if and only if the function $x \mapsto \mu(x)$ is computable for $x \in \Sigma^*$. This property does not imply that the number

$$1 - \mu(\infty) = \mu(\Sigma^*) = \sum_{x \in \Sigma^*} \mu(x)$$

is computable. ⌟

Let us allow, for a moment, measures μ that are not probability measures: they may not even be finite. Metric and computability can be extended to this case, the universal test $\mathbf{t}_\mu(x)$ can also be generalized. The Coding Theorem 1.6.5 and other considerations suggest the introduction of the following notation, for an arbitrary measure μ:

Definition 4.5.4 We define the *algorithmic entropy* of a point x with respect to measure μ as

$$H_\mu(x) = -\mathbf{d}_\mu(x) = -\log \mathbf{t}_\mu(x). \tag{4.5.2}$$

⌟

Then, with # defined as the counting measure over the discrete set Σ^* (that is, $\#(S) = |S|$), we have

$$H(x) \stackrel{+}{=} H_{\#}(x).$$

This allows viewing $H_\mu(x)$ as a generalization of description complexity.

The following theorem generalizes an earlier known theorem stating that over a discrete space, for a computable measure, entropy is within an additive constant the same as "average complexity": $\mathcal{H}(\mu) \stackrel{+}{=} \mu^x H(x)$.

Theorem 4.5.1 *Let μ be a probability measure. Then we have*

$$\mathcal{H}_\nu(\mu) \leq \mu^x H_\nu(x \mid \mu). \tag{4.5.3}$$

If X is a discrete space then the following estimate also holds:

$$\mathcal{H}_\nu(\mu) \stackrel{+}{>} \mu^x H_\nu(x \mid \mu). \tag{4.5.4}$$

Proof. Let δ be the measure with density $\mathbf{t}_\nu(x \mid \mu)$ with respect to ν: $\mathbf{t}_\nu(x \mid \mu) = \frac{\delta(dx)}{\nu(dx)}$. Then $\delta(X) \leq 1$. It is easy to see from the maximality property of $\mathbf{t}_\nu(x \mid \mu)$ that $\mathbf{t}_\nu(x \mid \mu) > 0$, therefore according to Proposition A.2.23, we have $\frac{\nu(dx)}{\delta(dx)} = \left(\frac{\delta(dx)}{\nu(dx)}\right)^{-1}$. Using Proposition A.2.23 and 4.5.2:

$$\begin{aligned}
\mathcal{H}_\nu(\mu) &= -\mu^x \log \frac{\mu(dx)}{\nu(dx)}, \\
-\mu^x H_\nu(x \mid \mu) &= \mu^x \log \frac{\delta(dx)}{\nu(dx)} = -\mu^x \log \frac{\nu(dx)}{\delta(dx)}, \\
\mathcal{H}_\nu(\mu) - \mu^x H_\nu(x \mid \mu) &= -\mu^x \log \frac{\mu(dx)}{\delta(dx)} \leq -\mu(X) \log \frac{\mu(X)}{\delta(X)} \leq 0.
\end{aligned}$$

This proves (4.5.3).

Over a discrete space X, the function $(x, \mu, \nu) \mapsto \frac{\mu(dx)}{\nu(dx)} = \frac{\mu(x)}{\nu(x)}$ is computable, therefore by the maximality property of $H_\nu(x \mid \mu)$ we have $\frac{\mu(dx)}{\nu(dx)} \stackrel{*}{<} \mathbf{t}_\nu(x \mid \mu)$, hence $\mathcal{H}_\nu(\mu) = -\mu^x \log \frac{\mu(dx)}{\nu(dx)} \stackrel{+}{>} \mu^x H_\nu(x \mid \mu)$. □

4.5.3 Addition theorem

The Addition Theorem (3.1.5) can be generalized to the algorithmic entropy $H_\mu(x)$ introduced in (4.5.2) (a somewhat similar generalization appeared in [52]). The generalization, defining $H_{\mu,\nu} = H_{\mu\times\nu}$, is

$$H_{\mu,\nu}(x, y) \stackrel{+}{=} H_\mu(x \mid \nu) + H_\nu(y \mid x,\ H_\mu(x \mid \nu),\ \mu). \tag{4.5.5}$$

Before proving the general addition theorem, we establish a few useful facts.

Proposition 4.5.5 *We have*

$$H_\mu(x \mid \nu) \stackrel{+}{<} -\log \nu^y 2^{-H_{\mu,\nu}(x,y)}.$$

Proof. The function $f(x, \mu, \nu)$ that is the right-hand side, is upper semicomputable by definition, and obeys $\mu^x 2^{-f(x,\mu,\nu)} \leq 1$. Therefore the inequality follows from the minimum property of $H_\mu(x)$. □

Let $z \in \mathbb{N}$, then the inequality

$$H_\mu(x) \stackrel{+}{<} H(z) + H_\mu(x \mid z) \tag{4.5.6}$$

will be a simple consequence of the general addition theorem. The following lemma, needed in the proof of the theorem, generalizes this inequality somewhat:

Lemma 4.5.6 *For a computable function $(y, z) \mapsto f(y, z)$ over $\mathbb{N}$, we have*

$$H_\mu(x \mid y) \stackrel{+}{<} H(z) + H_\mu(x \mid f(y, z)).$$

Proof. The function

$$(x, y, \mu) \mapsto g_\mu(x, y) = \sum_z 2^{-H_\mu(x \mid f(y,z)) - H(z)}$$

is lower semicomputable, and $\mu^x g_\mu(x, y) \leq \sum_z 2^{-H(z)} \leq 1$. Hence $g_\mu(x, y) \stackrel{*}{<} 2^{-H_\mu(x \mid y)}$. The left-hand side is a sum, hence the inequality holds for each element of the sum: just what we had to prove. □

The following monotonicity property will be needed:

Lemma 4.5.7 *For $i < j$ we have*

$$i + H_\mu(x \mid i) \stackrel{+}{<} j + H_\mu(x \mid j).$$

Proof. From Lemma 4.5.6, with $f(i, n) = i + n$ we have

$$H_\mu(x \mid i) - H_\mu(x \mid j) \stackrel{+}{<} H(j - i) \stackrel{+}{<} j - i.$$

□

Let us generalize the minimum property of $H_\mu(x)$.

Proposition 4.5.8 *Let $(y, \nu) \mapsto F_\nu(y)$ be an upper semicomputable function with values in $\overline{\mathbb{Z}} = \mathbb{Z} \cup \{-\infty, \infty\}$. Then by Corollary 4.1.3 among the lower semicomputable functions $(x, y, \nu) \mapsto g_\nu(x, y)$ with $\nu^x g_\nu(x, y) \leq 2^{-F_\nu(y)}$ there is one that is maximal to within a multiplicative constant. Choosing $f_\nu(x, y)$ as such a function we have for all x with $F_\nu(y) > -\infty$:*

$$f_\nu(x, y) \stackrel{*}{=} 2^{-F_\nu(y)} \mathbf{t}_\nu(x \mid y, F_\nu(y)),$$

or in logarithmic notation $-\log f_\nu(x, y) \stackrel{+}{=} F_\nu(y) + H_\nu(x \mid y, F_\nu(y))$.

Proof. To prove the inequality $\stackrel{*}{>}$, define

$$g_\nu(x, y, m) = \max_{i \geq m} 2^{-i} \mathbf{t}_\nu(x \mid y, i).$$

Function $g_\nu(x, y, m)$ is lower semicomputable and decreasing in m. Therefore

$$g_\nu(x, y) = g_\nu(x, y, F_\nu(y))$$

is also lower semicomputable since it is obtained by substituting an upper semicomputable function for m in $g_\nu(x, y, m)$. The multiplicative form of Lemma 4.5.7 implies

$$\begin{aligned} g_\nu(x, y, m) &\stackrel{*}{=} 2^{-m} \mathbf{t}_\nu(x \mid y, m), \\ g_\nu(x, y) &\stackrel{*}{=} 2^{-F_\nu(y)} \mathbf{t}_\nu(x \mid y, F_\nu(y)). \end{aligned}$$

We have, since $\mathbf{t}_\nu$ is a test:

$$\begin{aligned} \nu^x 2^{-m} \mathbf{t}_\nu(x \mid y, m) &\leq 2^{-m}, \\ \nu^x g_\nu(x, y) &\stackrel{*}{<} 2^{-F_\nu(y)}, \end{aligned}$$

implying $g_\nu(x, y) \stackrel{*}{<} f_\nu(x, y)$ by the optimality of $f_\nu(x, y)$.

To prove the upper bound, consider all lower semicomputable functions $\phi_e(x, y, m, \nu)$ ($e = 1, 2, \dots$). By Theorem 4.1.1, there is a recursive mapping $e \mapsto e'$ with the property that $\nu^x \phi_{e'}(x, y, m, \nu) \leq 2^{-m+1}$, and for each y, m, ν if $\nu^x \phi_e(x, y, m, \nu) < 2^{-m+1}$ then $\phi_e = \phi_{e'}$. Let us apply this transformation to the function $\phi_e(x, y, m, \nu) = f_\nu(x, y)$. The result is a lower semicomputable function $f'_\nu(x, y, m) = \phi_{e'}(x, y, m, \nu)$ with the property that $\nu^x f'_\nu(x, y, m) \leq 2^{-m+1}$, further $\nu^x f_\nu(x, y) \leq 2^{-m}$ implies $f'_\nu(x, y, m) = f_\nu(x, y)$. Now $(x, y, m, \nu) \mapsto 2^{m-1} f'_\nu(x, y, m)$ is a uniform test of x conditional on y, m and hence it is $\stackrel{*}{<} \mathbf{t}_\nu(x \mid y, m)$. Substituting $F_\nu(y)$ for m the relation $\nu^x f_\nu(x, y) \leq 2^{-m}$ is satisfied and hence we have

$$f_\nu(x, y) = f'_\nu(x, y, F_\nu(y)) \stackrel{*}{<} 2^{-F_\nu(y)+1} \mathbf{t}_\nu(x \mid y, F_\nu(y)).$$

□

As mentioned above, the theory generalizes to measures that are not probability measures. Taking $f_\mu(x, y) = 1$ in Proposition 4.5.8 gives the inequality

$$H_\mu(x \mid \lfloor \log \mu(X) \rfloor) \stackrel{+}{<} \log \mu(X),$$

with a physical meaning when μ is the phase space measure. Using (4.5.6), this implies

$$H_\mu(x) \stackrel{+}{<} \log \mu(X) + H(\lfloor \log \mu(X) \rfloor). \tag{4.5.7}$$

Theorem 4.5.2 (General addition) *The following inequality holds:*

$$H_{\mu,\nu}(x,y) \stackrel{+}{=} H_\mu(x \mid \nu) + H_\nu(y \mid x,\ H_\mu(x \mid \nu),\ \mu).$$

Proof. To prove the inequality $\stackrel{+}{<}$ define $G_{\mu,\nu}(x,y,m) = \min_{i \geq m}\ i + H_\nu(y \mid x, i, \mu)$. This function is upper semicomputable and increasing in m. Therefore function

$$G_{\mu,\nu}(x,y) = G_{\mu,\nu}(x,y,H_\mu(x \mid \nu))$$

is also upper semicomputable since it is obtained by substituting an upper semicomputable function for m in $G_{\mu,\nu}(x,y,m)$. Lemma 4.5.7 implies

$$G_{\mu,\nu}(x,y,m) \stackrel{+}{=} m + H_\nu(y \mid x, m, \mu),$$
$$G_{\mu,\nu}(x,y) \stackrel{+}{=} H_\mu(x \mid \nu) + H_\nu(y \mid x, H_\mu(x \mid \nu), \mu).$$

Now, we have

$$\nu^y 2^{-m - H_\nu(y \mid x,m,\mu)} \leq 2^{-m},$$
$$\nu^y 2^{-G_{\mu,\nu}(x,y)} \stackrel{*}{<} 2^{-H_\mu(x \mid \mu)}.$$

Integrating over x by μ gives $\mu^x \nu^y 2^{-G} \stackrel{*}{<} 1$, implying $H_{\mu,\nu}(x,y) \stackrel{+}{<} G_{\mu,\nu}(x,y)$ by the minimality property of $H_{\mu,\nu}(x,y)$. This proves the $\stackrel{+}{<}$ half of the theorem.

To prove the inequality $\stackrel{+}{>}$ let $f_\nu(x,y,\mu) = 2^{-H_{\mu,\nu}(x,y)}$. Proposition 4.5.5 implies that there is a constant c with $\nu^y f_\nu(x,y,\mu) \leq 2^{-H_\mu(x \mid \nu)+c}$. Let

$$F_\nu(x,\mu) = H_\mu(x \mid \nu).$$

Proposition 4.5.8 gives (substituting y for x and (x,μ) for y):

$$H_{\mu,\nu}(x,y) = -\log f_\nu(x,y,\mu) \stackrel{+}{>} F_\nu(x,\mu) + H_\nu(y \mid x, F_\nu(x,\mu), \mu),$$

which is what needed to be proved. □

The function $H_\mu(\omega)$ behaves quite differently for different kinds of measures μ. Recall the following property of complexity:

$$H(f(x) \mid y) \stackrel{+}{<} H(x \mid g(y)) \stackrel{+}{<} H(x). \tag{4.5.8}$$

for any computable functions f, g. This implies

$$H(y) \stackrel{+}{<} H(x,y).$$

In contrast, if μ is a probability measure then

$$H_\nu(y) \stackrel{+}{>} H_{\mu,\nu}(\omega,y).$$

This comes from the fact that $2^{-H_\nu(y)}$ is a test for $\mu \times \nu$.

4.5.4 Information

Mutual information has been defined in Definition 3.1.9 as $I^*(x : y) = H(x) + H(y) - H(x, y)$. By the Addition theorem, we have $I^*(x : y) \stackrel{+}{=} H(y) - H(y \mid x, H(x)) \stackrel{+}{=} H(x) - H(x \mid y, H(y))$. The two latter expressions show that in some sense, $I^*(x : y)$ is the information held in x about y as well as the information held in y about x. (The terms $H(x)$, $H(y)$ in the conditions are logarithmic-sized corrections to this idea.) Using (4.3.5), it is interesting to view mutual information $I^*(x : y)$ as a deficiency of randomness of the pair (x, y) in terms of the expression $\overline{\mathbf{d}}_\mu$, with respect to $\mathbf{m} \times \mathbf{m}$:

$$I^*(x : y) = H(x) + H(y) - H(x, y) = \overline{\mathbf{d}}_{\mathbf{m}\times\mathbf{m}}(x, y).$$

Taking $\mathbf{m}$ as a kind of "neutral" probability, even if it is not quite such, allows us to view $I^*(x : y)$ as a "deficiency of independence". Is it also true that $I^*(x : y) \stackrel{+}{=} \mathbf{d}_{\mathbf{m}\times\mathbf{m}}(x)$? This would allow us to deduce, as Levin did, "information conservation" laws from randomness conservation laws.[2]

Expression $\mathbf{d}_{\mathbf{m}\times\mathbf{m}}(x)$ must be understood again in the sense of compactification, as in Section 4.3. There seem to be two reasonable ways to compactify the space $\mathbb{N} \times \mathbb{N}$: we either compactify it directly, by adding a symbol ∞, or we form the product $\overline{\mathbb{N}} \times \overline{\mathbb{N}}$. With either of them, preserving Theorem 4.6.1, we would have to check whether $H(x, y \mid \mathbf{m} \times \mathbf{m}) \stackrel{+}{=} H(x, y)$. But, knowing the function $\mathbf{m}(x) \times \mathbf{m}(y)$ we know the function $x \mapsto \mathbf{m}(x) \stackrel{*}{=} \mathbf{m}(x) \times \mathbf{m}(0)$, hence also the function $(x, y) \mapsto \mathbf{m}(x, y) = \mathbf{m}(\langle x, y\rangle)$. Using this knowledge, it is possible to develop an argument similar to the proof of Theorem 4.3.2, showing that $H(x, y \mid \mathbf{m} \times \mathbf{m}) \stackrel{+}{=} H(x, y)$ does not hold.

Question 1 *Is there a neutral measure M with the property that $I^*(x : y) = \mathbf{d}_{M\times M}(x, y)$? Is this true maybe for all neutral measures M? If not, how far apart are the expressions $\mathbf{d}_{M\times M}(x, y)$ and $I^*(x : y)$ from each other?*

4.6 Randomness and complexity

We have seen in the discrete case that complexity and randomness are closely related. The connection is more delicate technically in the continuous case, but its exploration led to some nice results.

[2] We cannot use the test $\bar{\mathbf{t}}_\mu$ for this, since it can be shown easily that it does not to obey randomness conservation.

4.6.1 Discrete space

It is known that for computable μ, the test $\mathbf{d}_\mu(x)$ can be expressed in terms of the description complexity of x (we will prove these expressions below). Assume that $\mathbf{X}$ is the (discrete) space of all binary strings. Then we have

$$\mathbf{d}_\mu(x) = -\log \mu(x) - H(x) + O(H(\mu)). \tag{4.6.1}$$

The meaning of this equation is the following. The expression $-\log \mu(x)$ is an upper bound (within $O(H(\mu))$) of the complexity $H(x)$, and nonrandomness of x is measured by the difference between the complexity and this upper bound. Assume that $\mathbf{X}$ is the space of infinite binary sequences. Then equation (4.6.1) must be replaced with

$$\mathbf{d}_\mu(x) = \sup_n \left(-\log \mu(x^{\leq n}) - H(x^{\leq n}) + O(H(\mu))\right). \tag{4.6.2}$$

For noncomputable measures, we cannot replace $O(H(\mu))$ in these relations with anything finite, as shown in the following example. Therefore however attractive and simple, $\exp(-\log \mu(x) - H(x))$ is not a universal uniform test of randomness.

Proposition 4.6.1 *There is a measure μ over the discrete space $\mathbf{X}$ of binary strings such that for each n, there is an x with $\mathbf{d}_\mu(x) = n - H(n)$ and $-\log \mu(x) - H(x) \stackrel{+}{<} 0$.*

Proof. Let us treat the domain of our measure μ as a set of pairs (x, y). Let $x_n = 0^n$, for $n = 1, 2, \ldots$. For each n, let y_n be some binary string of length n with the property $H(x_n, y_n) > n$. Let $\mu(x_n, y_n) = 2^{-n}$. Then $-\log \mu(x_n, y_n) - H(x_n, y_n) \leq n - n = 0$. On the other hand, let $t_\mu(x, y)$ be the test nonzero only on strings x of the form x_n:

$$t_\mu(x_n, y) = \frac{\mathbf{m}(n)}{\sum_{z \in \mathcal{B}^n} \mu(x_n, z)}.$$

The form of the definition ensures semicomputability and we also have

$$\sum_{x,y} \mu(x, y) t_\mu(x, y) \leq \sum_n \mathbf{m}(n) < 1,$$

therefore t_μ is indeed a test. Hence $\mathbf{t}_\mu(x, y) \stackrel{*}{>} t_\mu(x, y)$. Taking logarithms, $\mathbf{d}_\mu(x_n, y_n) \stackrel{+}{>} n - H(n)$. □

The same example shows that the test defined as $\exp(-\log \mu(x) - H(x))$ over discrete sets, does not satisfy the randomness conservation property.

Proposition 4.6.2 *The test defined as $f_\mu(x) = \exp(-\log \mu(x) - H(x))$ over discrete spaces $\mathbf{X}$ does not obey the conservation of randomness.*

Proof. Let us use the example of Proposition 4.6.1. Consider the function $\pi : (x, y) \mapsto x$. The image of the measure μ under the projection is $(\pi\mu)(x) = \sum_y \mu(x, y)$. Thus, $(\pi\mu)(x_n) = \mu(x_n, y_n) = 2^{-n}$. Then we have seen that $\log f_\mu(x_n, y_n) \leq 0$. On the other hand,

$$\log f_{\pi\mu}(\pi(x_n, y_n)) = -\log(\pi\mu)(x_n) - H(x_n) \stackrel{+}{=} n - H(n).$$

Thus, the projection π takes a random pair (x_n, y_n) into an object x_n that is very nonrandom (when randomness is measured using the tests f_μ). □

In the example, we have the abnormal situation that a pair is random but one of its elements is nonrandom. Therefore even if we would not insist on universality, the test $\exp(-\log \mu(x) - H(x))$ is unsatisfactory.

Looking into the reasons of the nonconservation in the example, we will notice that it could only have happened because the test f_μ is too special. The fact that $-\log(\pi\mu)(x_n) - H(x_n)$ is large should show that the pair (x_n, y_n) can be enclosed into the "simple" set $\{x_n\} \times \mathbf{Y}$ of small probability; unfortunately, this observation does not reflect on $-\log \mu(x, y) - H(x, y)$ (it does for computable μ).

It is a natural idea to modify equation (4.6.1) in such a way that the complexity $H(x)$ is replaced with $H(x \mid \mu)$. However, this expression must be understood properly. We need to use the definition of $H(x)$ as $-\log \mathbf{m}(x)$ directly, and not as prefix complexity.

Let us mention the following easy fact:

Proposition 4.6.3 *If μ is a computable measure then $H(x \mid \mu) \stackrel{+}{=} H(x)$. The constant in $\stackrel{+}{=}$ depends on the description complexity of μ.*

Theorem 4.6.1 *If $\mathbf{X}$ is the discrete space Σ^* then we have*

$$\mathbf{d}_\mu(x) \stackrel{+}{=} -\log \mu(x) - H(x \mid \mu). \tag{4.6.3}$$

Note that in terms of the algorithmic entropy notation introduced in (4.5.2), this theorem can be expressed as

$$H_\mu(x) \stackrel{+}{=} H(x \mid \mu) + \log \mu(x).$$

Proof. In exponential notation, equation (4.6.3) can be written as $\mathbf{t}_\mu(x) \stackrel{*}{=} \mathbf{m}(x \mid \mu)/\mu(x)$. Let us prove $\stackrel{*}{>}$ first. We will show that the right-hand side of this inequality is a test, and hence $\stackrel{*}{<} \mathbf{t}_\mu(x)$. However, the right-hand side is clearly lower semicomputable in (x, μ) and when we "integrate" it (multiply it by $\mu(x)$ and sum it), its sum is ≤ 1; thus, it is a test.

Let us prove $\stackrel{*}{<}$ now. The expression $\mathbf{t}_\mu(x)\mu(x)$ is clearly lower semicomputable in (x, μ), and its sum is ≤ 1. Hence, it is $\stackrel{+}{<} \mathbf{m}(x \mid \mu)$. □

Remark 4.6.4 It important not to consider relative computation with respect to μ as *oracle computation* in the ordinary sense. Theorem 4.3.1 below will show the existence of a measure with respect to which every element is random. If randomness is defined using μ as an oracle then we can always find elements nonrandom with respect to μ.

For similar reasons, the proof of the Coding Theorem does not transfer to the function $H(x \mid \mu)$ since an interpreter function should have the property of *intensionality*, depending only on μ and not on the sequence representing it. (It does transfer without problem to an oracle version of $H^{\mu}(x)$.) The Coding Theorem still may hold, at least in some cases: this is currently not known. Until we know this, we cannot interpret $H(x \mid \mu)$ as description complexity in terms of interpreters and codes.

(Thanks to Alexander Shen for this observation: this remark corrects an error in the paper [21].) ⌟

4.6.2 Non-discrete spaces

For non-discrete spaces, unfortunately, we can only provide a less intuitive expression.

Proposition 4.6.5 *let $\mathbf{X} = (X, d, D, \alpha)$ be a complete computable metric space, and let $\mathcal{E}$ be the enumerated set of bounded Lipschitz functions introduced in (A.1.3), but for the space $\mathbf{M}(\mathbf{X}) \times \mathbf{X}$. The uniform test of randomness $\mathbf{t}_{\mu}(x)$, can be expressed as*

$$\mathbf{t}_{\mu}(x) \stackrel{*}{=} \sum_{f \in \mathcal{E}} f(\mu, x) \frac{\mathbf{m}(f \mid \mu)}{\mu^{y} f(\mu, y)}. \tag{4.6.4}$$

Proof. For $\stackrel{*}{>}$, we will show that the right-hand side of the inequality is a test, and hence $\stackrel{*}{<} \mathbf{t}_{\mu}(x)$. For simplicity, we skip the notation about the enumeration of $\mathcal{E}$ and treat each element f as its own name. Each term of the sum is clearly lower semicomputable in (f, x, μ), hence the sum is lower semicomputable in (x, μ). It remains to show that the μ-integral of the sum is ≤ 1. But, the μ-integral of the generic term is $\leq \mathbf{m}(f \mid \mu)$, and the sum of these terms is ≤ 1 by the definition of the function $\mathbf{m}(\cdot \mid \cdot)$. Thus, the sum is a test.

For $\stackrel{*}{<}$, note that $(\mu, x) \mapsto \mathbf{t}_{\mu}(x)$, as a lower semicomputable function, is the supremum of functions in $\mathcal{E}$. Denoting their differences by $f_i(\mu, x)$, we have $\mathbf{t}_{\mu}(x) = \sum_i f_i(\mu, x)$. The test property implies $\sum_i \mu^x f_i(\mu, x) \leq 1$. Since the function $(\mu, i) \mapsto \mu^x f_i(\mu, x)$ is lower semicomputable, this implies $\mu^x f_i(\mu, x) \stackrel{*}{<} \mathbf{m}(i \mid \mu)$, and hence

$$f_i(\mu, x) \stackrel{*}{<} f_i(\mu, x) \frac{\mathbf{m}(i \mid \mu)}{\mu^x f_i(\mu, x)}.$$

It is easy to see that for each $f \in \mathcal{E}$ we have

$$\sum_{i:f_i=f} \mathbf{m}(i \mid \mu) \leq \mu(f \mid \mu),$$

which leads to (4.6.4). □

Remark 4.6.6 If we only want the $\overset{*}{>}$ part of the result, then $\mathcal{E}$ can be replaced with any enumerated computable sequence of bounded computable functions. ⌟

4.6.3 Infinite sequences

In case of the space of infinite sequences and a computable measure, Theorem 2.3.4 gives a characterization of randomness in terms of complexity. This theorem does not seem to transfer to a more general situation, but under some conditions, at least parts of it can be extended.

For arbitrary measures and spaces, we can say a little less:

Proposition 4.6.7 *For all measures $\mu \in \mathcal{M}_R(X)$, for the deficiency of randomness $\mathbf{d}_\mu(x)$, we have*

$$\mathbf{d}_\mu(x) \overset{+}{>} \sup_n \big(-\log \overline{\mu}(x^{\leq n}) - H(x^{\leq n} \mid \mu)\big). \tag{4.6.5}$$

Proof. Consider the function

$$f_\mu(x) = \sum_s 1_{\Gamma_s}(x) \frac{\mathbf{m}(s \mid \mu)}{\overline{\mu}(\Gamma_s)} = \sum_n \frac{\mathbf{m}(x^{\leq n} \mid \mu)}{\overline{\mu}(x^{\leq n})} \geq \sup_n \frac{\mathbf{m}(x^{\leq n} \mid \mu)}{\overline{\mu}(x^{\leq n})}.$$

The function $(\mu, x) \mapsto f_\mu(x)$ is clearly lower semicomputable and satisfies $\mu^x f_\mu(x) \leq 1$, and hence

$$\mathbf{d}_\mu(x) \overset{+}{>} \log f(x) \overset{+}{>} \sup_n \big(-\log \overline{\mu}(x^{\leq n}) - H(x^{\leq n} \mid \mu)\big).$$

□

Definition 4.6.8 Let $\mathcal{M}_R(X)$ be the set of measures μ with $\mu(X) = R$. ⌟

We will be able to prove the $\overset{+}{>}$ part of the statement of Theorem 2.3.4 in a more general space, and without assuming computability. Assume that a separating sequence $b_1, b_2, \ldots$ is given as defined in Subsection 4.7, along with the set X^0. For each $x \in X^0$, the binary sequence $x_1, x_2, \ldots$ has been defined. Let

$$\overline{\mu}(\Gamma_s) = R - \sum \{\mu(\Gamma_{s'}) : l(s) = l(s'),\ s' \neq s\}.$$

Then $(s, \mu) \mapsto \mu(\Gamma_s)$ is lower semicomputable, and $(s, \mu) \mapsto \overline{\mu}(\Gamma_s)$ is upper semicomputable. And, every time that the functions $b_i(x)$ form a partition with μ-continuity, we have $\overline{\mu}(\Gamma_s) = \mu(\Gamma_s)$ for all s.

Theorem 4.6.2 *Suppose that the space X is effectively compact. Then for all computable measures $\mu \in \mathcal{M}^0_R(X)$, for the deficiency of randomness $\mathbf{d}_\mu(x)$, the characterization* (2.3.3) *holds.*

Proof. The proof of part $\stackrel{+}{>}$ of the inequality follows directly from Proposition 4.6.7, just as in the proof of Theorem 2.3.4.

The proof of $\stackrel{+}{<}$ is also similar to the proof of that theorem. The only part that needs to be reproved is the statement that for every lower semicomputable function f over X, there are computable sequences $y_i \in \mathbb{N}^*$ and $q_i \in \mathbb{Q}$ with $f(x) = \sup_i q_i 1_{y_i}(x)$. This follows now, since according to Proposition 4.7.7, the cells Γ_y form a basis of the space X. □

4.6.4 Bernoulli tests

In this part, we will give characterize Bernoulli sequences in terms of their complexity growth.

Recall Definition 4.2.11: A function $f : \mathbb{B}^* \to \mathbb{R}$ is a combinatorial Bernoulli test if

a) It is lower semicomputable.

b) It is monotonic with respect to the prefix relation.

c) For all $0 \le k \le n$ we have $\sum_{x \in \mathbb{B}(n,k)} f(x) \le \binom{n}{k}$.

According to Theorem 4.2.3, a sequence ξ is nonrandom with respect to all Bernoulli measures if and only if $\sup_n b(\xi^{\le n}) = \infty$, where $b(x)$ is a combinatorial Bernoulli test.

We need some definitions.

Definition 4.6.9 For a finite or infinite sequence x let $S_n(x) = \sum_{i=1}^n x(i)$.

For $0 \le p \le 1$ and integers $0 \le k \le n$, denote $B_p(n,k) = \binom{n}{k} p^k (1-p)^{n-k}$.

An upper semicomputable function $D : \mathbb{N}^2 \to \mathbb{N}$, defined for $n \ge 1$, $0 \le k \le n$ will be called a *gap function* if

$$\sum_{n \ge 1} \sum_{k=0}^{n} B_p(n,k) 2^{-D(n,k)} \le 1 \tag{4.6.6}$$

holds for all $0 \le p \le 1$. A gap function $D(n,k)$ is *optimal* if for every other gap function $D'(n,k)$ there is a $c_{D'}$ with $D(n,k) \le D'(n,k) + c_{D'}$. ⌟

Proposition 4.6.10 *There is an optimal gap function $D(n,k) \stackrel{+}{<} H(n)$.*

Proof. The existence is proved using the technique of Theorem 4.1.1. For the inequality it is sufficient to note that $H(n)$ is a gap function. Indeed, we have

$$\sum_{n\geq 1}\sum_{k=0}^{n} B_p(n,k)2^{-H(n)} = \sum_{n\geq 1} 2^{-H(n)} \leq 1. \qquad \square$$

Definition 4.6.11 Let us fix an optimal gap function and denote it by $\Delta(n,k)$. ⌟

Now we can state the test characterization theorem for Bernoulli tests.

Theorem 4.6.3 *Denoting by $b(\xi)$ the universal class test for the Bernoulli sequences, we have $b(\xi) \stackrel{*}{=} \overline{b}(\xi)$, where*

$$\log \overline{b}(\xi) = \sup_n \log \binom{n}{k} - H(\xi^{\leq n} \mid n, k, \Delta(n,k)) - \Delta(n,k),$$

with $k = S_n(\xi)$.

Proof. Let $\delta(n,k) = 2^{-\Delta(n,k)}$.

Claim 4.6.12 *Consider lower semicomputable functions $\gamma : \mathbb{B}^* \to \mathbb{R}_+$ such that for all n, k we have*

$$\sum_{y\in\mathbb{B}(n,k)} \gamma(y) \leq 2^{-\Delta(n,k)}.$$

Among these functions there is one that is optimal (maximal to within a multiplicative constant). Calling it $\delta(y)$ we have

$$\delta(y) \stackrel{*}{=} \delta(n,k)\cdot \mathbf{m}(y \mid n, k, \Delta(n,k)).$$

Thus, the right-hand side is equal, within a multiplicative constant, to a lower semicomputable function of y.

Proof. This follows immediately from Proposition 4.5.8. with $\nu = \#$ (the counting measure) over the sets $\mathbb{B}(n,k)$. $\square$

Let us show $\overline{b}(\xi) \stackrel{*}{<} b(\xi)$. We have with $k = S_n(\xi)$ in the first line:

$$\begin{aligned}
\overline{b}(\xi) &= \sup_{n\geq 1} \binom{n}{k} \mathbf{m}(\xi^{\leq n} \mid n, k, \Delta(n,k))\delta(n,k) \\
&= \sup_{n\geq 1} \sum_{k=0}^{n} \binom{n}{k} \delta(n,k) \sum_{y\in\mathbb{B}(n,k)} 1_y(\xi)\mathbf{m}(y \mid n, k, \Delta(n,k)) \\
&\leq \sum_{n\geq 1} \sum_{k=0}^{n} \binom{n}{k} \sum_{y\in\mathbb{B}(n,k)} 1_y(\xi)\delta(n,k)\mathbf{m}(y \mid n, k, \Delta(n,k)) \\
&\stackrel{*}{=} \sum_{n\geq 1} \sum_{k=0}^{n} \binom{n}{k} \sum_{y\in\mathbb{B}(n,k)} 1_y(\xi)\delta(y),
\end{aligned}$$

using the notation of Claim 4.6.12 above. Let $t(\xi)$ denote the right-hand side here, which is thus a lower semicomputable function. We have for all p:

$$B_p^{\xi} t(\xi) \stackrel{*}{=} \sum_{n\geq 1} \sum_{k=0}^{n} B_p(n,k)\delta(n,k) \sum_{y\in\mathbb{B}(n,k)} \mathbf{m}(y \mid n, k, \Delta(n,k)) \leq 1,$$

so $t(\xi) \stackrel{*}{>} \overline{b}(\xi)$ is a Bernoulli test, showing $\overline{b}(\xi) \stackrel{*}{<} b(\xi)$.

To show $b(\xi) \stackrel{*}{<} \overline{b}(\xi)$ we will follow the method of the proof of Theorem 2.3.4. Replace $b(\xi)$ with a rougher version:

$$b'(\xi) = \frac{1}{2} \max\{2^n : 2^n < b(\xi)\},$$

then we have $2b' < b$. There are computable sequences $y_i \in \mathbb{B}^*$ and $k_i \in \mathbb{N}$ with $b'(\xi) = \sup_i 2^{k_i} 1_{y_i}(\xi)$ with the property that if $i < j$ and $1_{y_i}(\xi) = 1_{y_j}(\xi) = 1$ then $k_i < k_j$. As in the imitated proof, we have $2b'(\xi) \geq \sum_i 2^{k_i} 1_{y_i}(\xi)$. The function $\gamma(y) = \sum_{y_i = y} 2^{k_i}$ is lower semicomputable. We have

$$b \geq 2b'(\xi) \geq \sum_i 2^{k_i} 1_{y_i}(\xi) = \sum_{y\in\mathbb{N}^*} 1_y(\xi)\gamma(y) = \sum_n \sum_{k=0}^{n} \sum_{y\in\mathbb{B}(n,k)} 1_y(\xi)\gamma(y). \quad (4.6.7)$$

By Theorem 4.2.3 we can assume $\sum_{y\in\mathbb{B}(n,k)} \gamma(y) \leq \binom{n}{k}$. Let

$$\begin{aligned}
\delta'(y) &= \gamma(y)\binom{n}{k}^{-1} \leq 1, \\
\delta'(n,k) &= \sum_{y\in\mathbb{B}(n,k)} \delta'(y).
\end{aligned}$$

Since $1 \geq B_p b \geq B_p(2b')$ for all p, we have

$$1 \geq \sum_n \sum_{k=0}^{n} \sum_{y \in \mathbb{B}(n,k)} \gamma(y) B_p^{\xi} 1_y(\xi) = \sum_n \sum_{k=0}^{n} p^k (1-p)^{n-k} \sum_{y \in \mathbb{B}(n,k)} \gamma(y)$$
$$= \sum_n \sum_{k=0}^{n} B_p(n,k) \sum_{y \in \mathbb{B}(n,k)} \delta'(y) = \sum_n \sum_{k=0}^{n} B_p(n,k) \delta'(n,k).$$

Thus $\delta'(n,k)$ is a gap function, hence $\delta'(n,k) \stackrel{*}{<} 2^{-\Delta(n,k)}$, and by Claim 4.6.12 we have

$$\gamma(y) \binom{n}{k}^{-1} = \delta'(y) \stackrel{*}{<} \delta(y) \stackrel{*}{=} \delta(n,k) \cdot \mathbf{m}(y \mid n, k, \Delta(n,k)).$$

Substituting back into (4.6.7) finishes the proof of $b(\xi) \stackrel{*}{<} \bar{b}(\xi)$. □

4.7 Cells

This section allows to transfer some of the results on sequence spaces to more general spaces, by encoding the elements into sequences. The reader who is only interested in sequences can skip this section.

4.7.1 Partitions

The coordinates of the sequences into which we want to encode our elements will be obtained via certain partitions.

Recall from Definition A.2.33 that a measurable set A is said to be a *μ-continuity set* if $\mu(\partial A) = 0$ where ∂A is the boundary of A.

Definition 4.7.1 (Continuity partition) Let $f : X \to \mathbb{R}$ be a bounded computable function, and let $\alpha_1 < \cdots < \alpha_k$ be rational numbers, and let μ be a computable measure with the property that $\mu f^{-1}(\alpha_j) = 0$ for all j.

In this case, we will say that α_j are *μ-continuity points* of f. Let $\alpha_0 = -\infty$, $\alpha_{k+1} = \infty$, and for $j = 0, \ldots, k$, let Let $U_j = f^{-1}((j, j+1))$. The sequence of disjoint computably enumerable open sets $(U_0, \ldots, U_k)$ will be called the *partition generated by* $f, \alpha_1, \ldots, \alpha_k$.

If we have several partitions $(U_{i0}, \ldots, U_{i,k})$, generated by different functions f_i $(i = 1, \ldots, m)$ and cutoff sequences $(\alpha_{ij} : j = 1, \ldots, k_i)$ made up of μ-continuity points of f_i then we can form a new partition generated by all possible intersections

$$V_{j_1,\ldots,j_n} = U_{1,j_1} \cap \cdots \cap U_{m,j_m}.$$

A partition of this kind will be called a *continuity partition*. The sets $V_{j_1,\dots,j_n}$ will be called the *cells* of this partition. ⌟

The following is worth noting.

Proposition 4.7.2 *In a partition as given above, the values $\mu V_{j_1,\dots,j_n}$ are computable from the names of the functions f_i and the cutoff points α_{ij}.*

Proof. Straightforward. □

Let us proceed to defining cells.

Definition 4.7.3 (Separating sequence) Assume that a computable sequence of functions $b_1(x), b_2(x), \dots$ over X is given, with the property that for every pair $x_1, x_2 \in X$ with $x_1 \neq x_2$, there is a j with $b_j(x_1) \cdot b_j(x_2) < 0$. Such a sequence will be called a *separating sequence*. Let us give the correspondence between the set $\mathbb{B}^{\mathbb{N}}$ of infinite binary sequences and elements of the set

$$X^0 = \{x \in X : b_j(x) \neq 0,\ \ j = 1, 2, \dots\}.$$

For a binary string $s_1 \cdots s_n = s \in \mathbb{B}^*$, let

$$\Gamma_s$$

be the set of elements of X with the property that for $j = 1, \dots, n$, if $s_j = 0$ then $b_j(\omega) < 0$, otherwise $b_j(\omega) > 0$.

The separating sequence will be called μ-*continuity* sequence if $\mu(X^0) = 0$. ⌟

This correspondence has the following properties.

a) $\Gamma_\Lambda = X$.

b) For each $s \in \mathbb{B}$, the sets Γ_{s0} and Γ_{s1} are disjoint and their union is contained in Γ_s.

c) For $x \in X^0$, we have $\{x\} = \bigcap_{x \in \Gamma_s} \Gamma_s$.

Definition 4.7.4 (Cells) If string s has length n then Γ_s will be called a *canonical n-cell*, or simply canonical cell, or n-cell. From now on, whenever Γ denotes a subset of X, it means a canonical cell. We will also use the notation

$$l(\Gamma_s) = l(s).$$

⌟

The three properties above say that if we restrict ourselves to the set X^0 then the canonical cells behave somewhat like binary subintervals: they divide X^0 in

half, then each half again in half, etc. Moreover, around each point, these canonical cells become "arbitrarily small", in some sense (though, they may not be a basis of neighborhoods). It is easy to see that if Γ_{s_1}, Γ_{s_2} are two canonical cells then they either are disjoint or one of them contains the other. If $\Gamma_{s_1} \subset \Gamma_{s_2}$ then s_2 is a prefix of s_1. If, for a moment, we write $\Gamma_s^0 = \Gamma_s \cap X^0$ then we have the disjoint union $\Gamma_s^0 = \Gamma_{s0}^0 \cup \Gamma_{s1}^0$.

Let us use the following notation.

Definition 4.7.5 For an n-element binary string s, for $x \in \Gamma_s$, we will write

$$s = x^{\leq n} = x_1 \cdots x_n, \qquad \mu(s) = \mu(\Gamma_s).$$

The 2^n cells (some of them possibly empty) of the form Γ_s for $l(s) = n$ form a partition

$$\mathcal{P}_n$$

of X^0. ⌟

Thus, for elements of X^0, we can talk about the n-th bit x_n of the description of x: it is uniquely determined.

Examples 4.7.6

1. If $\mathbf{X}$ is the set of infinite binary sequences with its usual topology, the functions $b_n(x) = x_n - 1/2$ generate the usual cells, and $\mathbf{X}^0 = \mathbf{X}$.
2. If $\mathbf{X}$ is the interval $[0, 1]$, let $b_n(x) = -\sin(2^n \pi x)$. Then cells are open intervals of the form $(k \cdot 2^{-n}, (k + 1) \cdot 2^n)$, the correspondence between infinite binary strings and elements of X^0 is just the usual representation of x as the binary decimal string $0.x_1x_2 \ldots$.

⌟

When we fix canonical cells, we will generally assume that the partition chosen is also "natural". The bits $x_1, x_2, \ldots$ could contain information about the point x in decreasing order of importance from a macroscopic point of view. For example, for a container of gas, the first few bits may describe, to a reasonable degree of precision, the amount of gas in the left half of the container, the next few bits may describe the amounts in each quarter, the next few bits may describe the temperature in each half, the next few bits may describe again the amount of gas in each half, but now to more precision, etc. From now on, whenever Γ denotes a subset of X, it means a canonical cell. From now on, for elements of X^0, we can talk about the n-th bit x_n of the description of x: it is uniquely determined.

The following observation will prove useful.

Proposition 4.7.7 *Suppose that the space* **X** *is effectively compact*[3] *and we have a separating sequence* $b_i(x)$ *as given above. Then the cells* Γ_s *form a basis of the space* **X**.

Proof. We need to prove that for every ball $B(x, r)$, the there is a cell $x \in \Gamma_s \subset B(x, r)$. Let C be the complement of $B(x, r)$. For each point y of C, there is an i such that $b_i(x) \cdot b_i(y) < 0$. In this case, let $J^0 = \{z : b_i(z) < 0\}$, $J^1 = \{z : b_i(z) > 0\}$. Let $J(y) = J^p$ such that $y \in J^p$. Then $C \subset \bigcup_y J(y)$, and compactness implies that there is a finite sequence $y_1, \ldots, y_k$ with $C \subset \bigcup_{j=1}^{k} J(y_j)$. Clearly, there is a cell

$$x \in \Gamma_s \subset B(x, r) \setminus \bigcup_{j=1}^{k} J(y_j).$$

□

4.7.2 Computable probability spaces

If a separating sequence is given in advance, we may restrict attention to the class of measures that make our sequence a μ-continuity sequence:

Definition 4.7.8 Let $\mathcal{M}^0(X)$ be the set of those probability measures μ in $\mathcal{M}(X)$ for which $\mu(X \setminus X^0) = 0$. ⌟

On the other hand, for each computable measure μ, a computable separating sequence can be constructed that is a μ-continuity sequence. Recall that $B(x, r)$ is the ball of center x and radius r. Let $D = \{s_1, s_2, \ldots\} = \{\alpha(1), \alpha(2), \ldots\}$ be the canonical enumeration of the canonical dense set D.

Theorem 4.7.1 (Hoyrup-Rojas) *There is a sequence of uniformly computable reals* $(r_n)_{n\in\mathbb{N}}$ *such that* $(B(s_i, r_n))_{i,n}$ *is a basis of balls that are* μ*-continuity sets. This basis is constructively equivalent to the original one, consisting of all balls* $B(s_i, r)$, $r \in \mathbb{Q}$.

Corollary 4.7.9 *There is a computable separating sequence with the* μ*-continuity property.*

Proof. Let us list all balls $B(s_i, r_n)$ into a single sequence $B(s_{i_k}, r_{n_k})$. The functions

$$b_k(x) = d(s_{i_k}, x) - r_{n_k}$$

give rise to the desired sequence. □

For the proof of the theorem, we use some preparation. Recall from Definition A.2.5 that an atom is a point with positive measure.

[3] It was noted by Hoyrup and Rojas that the qualification "effectively" is necessary here.

Lemma 4.7.10 *Let X be $\mathbb{R}$ or $\mathbb{R}^+$ or $[0,1]$. Let μ be a computable probability measure on X. Then there is a sequence of uniformly computable reals $(x_n)_n$ which is dense in X and contains no atoms of μ.*

Proof. Let I be a closed rational interval. We construct $x \in I$ with $\mu(\{x\}) = 0$. To do this, we construct inductively a nested sequence of closed intervals J_k of measure $< 2^{-k+1}$, with $J_0 = I$. Suppose $J_k = [a, b]$ has been constructed, with $\mu(J_k) < 2^{-k+1}$. Let $m = (b - a)/3$: one of the intervals $[a, a + m]$ and $[b - m, b]$ must have measure $< 2^{-k}$, and we can find it effectively—let it be J_{k+1}.

From a constructive enumeration $(I_n)_n$ of all the dyadic intervals, we can construct $x_n \in I_n$ uniformly. □

Corollary 4.7.11 *Let $(\mathbf{X}, \mu)$ be a computable metric space with a computable measure and let $(f_i)_i$ be a sequence of uniformly computable real valued functions on X. Then there is a sequence of uniformly computable reals $(x_n)_n$ that is dense in $\mathbb{R}$ and such that each x_n is a μ-continuity point of each f_i.*

Proof. Consider the uniformly computable measures $\mu_i = \mu \circ f_i^{-1}$ and define $\nu = \sum_i 2^{-i} \mu_i$. It is easy to see that ν is a computable measure and then, by Lemma 4.7.10, there is a sequence of uniformly computable reals $(x_n)_n$ which is dense in $\mathbb{R}$ and contains no atoms of ν. Since $\nu(A) = 0$ iff $\mu_i(A) = 0$ for all i, we get $\mu(\{f_i^{-1}(x_n)\}) = 0$ for all i, n. □

Proof of Theorem 4.7.1. Apply Corollary 4.7.11 to $f_i(x) = d(s_i, x)$.

Since every ball $B(s_i, r)$ can be expressed as a computably enumerable union of the balls of the type $B(s_i, r_n)$ just constructed, the two bases are constructively equivalent. □

5 Exercises and problems

Exercise 1 Define, for any two natural numbers r, s, a standard encoding cnv_s^r of base r strings x into base s strings with the property

$$|\mathrm{cnv}_s^r(x)| \leq |x|\frac{\log r}{\log s} + 1. \tag{5.0.1}$$

Solution. We will use $\mathcal{X} = N^N$, $\mathcal{X}_r = Z_r^N$ for the sets of infinite strings of natural numbers and r-ary digits respectively. For a sequence $p \in \mathcal{X}_r$, let $[p]_r$ denote the real number in the interval $[0, 1]$ which $0.p$ denotes in the base r number system. For p in $\mathbb{S}_r$, let $[p]_r = \{[pq]_r : q \in \mathcal{X}_r\}$.

For the r-ary string x, let v be the size of the largest s-ary intervals $[y]_s$ contained in the r-ary interval $[x]_r$. If $[z]_s$ is the leftmost among these intervals, then let $\mathrm{cnv}_s^r(x) = z$. This is a one-to-one encoding. We have $r^{-|x|} < 2sv$, since any $2s$ consecutive s-ary intervals of length v contain an s-ary interval of length sv. Therefore

$$|z| = -\log_s v < |x|\frac{\log r}{\log s} + 1 + \frac{\log 2}{\log s}$$

hence, since $2 \leq s$ and $|x|$ is integer, we have the inequality (5.0.1). □

Exercise 2 A function A from $\mathbb{S}_r \times \mathbb{S}$ to $\mathbb{S}$ is called an r-ary interpreter. Prove the following generalization of the Invariance Theorem. For any s, there is a p.r. s-ary interpreter U such that for any p.r. interpreter A there is a constant $c < \infty$ such that for all x, y we have

$$K_U(x \mid y) \leq K_A(x \mid y) + c. \tag{5.0.2}$$

Solution. Let $V : \mathbb{Z}_s^* \times \mathbb{Z}_s^* \times \mathbb{S} \to \mathbb{S}$ be a partial recursive function which is *universal*: such that for any p.r. s-ary interpreter A, there is a string a such that for all p, x, we have $A(p, x) = V(a, p, x)$.

The machine computing $U(p, x)$ tries to decompose p into $u^o v$ and outputs $V(u, v, x)$. Let us verify that U is optimal. Let A be a p.r. r-ary interpreter, B an

s-ary p.r. interpreter such that $B(\mathrm{cnv}_s^r(p), x) = A(p, x)$ for all p, x, a a binary string such that $B(p, x) = U(a, p, x)$ for all p, x. Let x, y be two strings. If $K_A(x \mid y) = \infty$, then the inequality (5.0.2) holds trivially. Otherwise, let p be a binary string of length $K_A(x \mid y)/\log r$ with $A(p, y) = x$. Then

$$U(a^o \mathrm{cnv}_s^r(p), y) = V(a, \mathrm{cnv}_s^r(p), y) = B(\mathrm{cnv}_s^r(p), y) = A(p, y) = x.$$

Since

$$|\mathrm{cnv}_s^r(p)| \leq |p| \log r/\log s + 1 = K_A(x \mid y)/\log s + 1,$$

we have

$$K_U(x \mid y) \leq (2|a| + K_A(x \mid y)/\log s + 1) \log s = K_A(x \mid y) + (2|a|) + 1) \log s.$$

□

Exercise 3 (S_n^m -theorem) Prove that there is a binary string b such that $U(p, q, x) = U(b^o p^o q, x)$ holds for all binary strings p, q and arbitrary strings x.

Exercise 4 (Schnorr) Notice that, apart from the conversion between representations, what our optimal interpreter does is the following. It treats the program p as a pair $(p(1), p(2))$ whose first member is the Gödel number of some interpreter for the universal p.r. function $V(p_1, p_2, x)$, and the second argument as a program for this interpreter. Prove that indeed, for any recursive pairing function $w(p, q)$, if there is a function f such that $|w(p, q)| \leq f(p) + |q|$ then w can be used to define an optimal interpreter.

Exercise 5 Refute the inequality $K(x, y) \stackrel{+}{<} K(x) + K(y)$.

Exercise 6 Prove the following sharpenings of Theorem 1.4.2.

$$K(x, y) \stackrel{+}{<} J(K(x)) + K(y \mid x, K(x)),$$
$$K(x, y) \stackrel{+}{<} K(x) + J(K(y \mid x, K(x))).$$

Prove Theorem 1.4.2 from here.

Exercise 7 (Schnorr) Prove $K(x + K(x)) \stackrel{+}{=} K(x)$. Can you generalize this result?

Exercise 8 (Schnorr) Prove that if $m < n$ then $m + K(m) \stackrel{+}{<} n + K(n)$.

Exercise 9 Prove

$$\log \binom{n}{k} \stackrel{+}{=} k \log \frac{n}{k} + (n - k) \log \frac{n}{n - k} + \frac{1}{2} \log \frac{n}{k(n - k)}.$$

Exercise 10 (Kamae) Prove that for any natural number k there is a string x such that for for all but finitely many strings y, we have

$$K(x) - K(x \mid y) \geq k.$$

In other words, there are some strings x such that almost any string y contains a large amount of information about them.

Solution. Strings x with this property are for example the ones which contain information obtainable with the help of any sufficiently large number. Let E be a r.e. set of integers. Let $e_0e_1e_2\ldots$ be the infinite string which is the characteristic function of E, and let $x(k) = e_0\ldots e_{2^k}$. We can suppose that $K(x(k)) \geq k$. Let $n_1, n_2, \ldots$ be a recursive enumeration of E without repetition, and let $\alpha(k) = \max\{i : n_i \leq 2^k\}$. Then for any number $t \geq \alpha(k)$ we have $K(x(k) \mid t) \stackrel{+}{<} \log k$. Indeed, a binary string of length k describes the number k. Knowing t we can enumerate $n_1, \ldots, n_t$ and thus learn $x(k)$. Therefore with any string y of length $\geq \alpha(k)$ we have $K(x) - K(x \mid y) \stackrel{+}{>} k - \log k$. □

Exercise 11 a) Prove that a real function f is computable iff there is a recursive function $g(x, n)$ with rational values, and $|f(x) - g(x, n)| < 1/n$.

b) Prove that a function f is semicomputable iff there exists a recursive function with rational values, (or, equivalently, a computable real function) $g(x, n)$ nondecreasing in n, with $f(x) = \lim_{n\to\infty} g(x, n)$.

Exercise 12 Prove that in Theorem 1.5.2 one can write "semicomputable" for "partial recursive".

Exercise 13 (Levin) Show that there is an upper semicomputable function $G(p, x, y)$ which for different finite binary strings p enumerates all upper semicomputable functions $F(x, y)$ satisfying the inequality (1.5.1). Prove

$$K(x \mid y) \stackrel{+}{=} \inf_p G(p, x, y) + J(|p|).$$

Exercise 14 Prove

$$\sum_{p\in\mathbb{B}^n} \mathbf{m}(p) \stackrel{*}{=} \mathbf{m}(n).$$

Exercise 15 (Solovay) Show that we cannot find effectively infinitely many places where some recursive upper bound of $H(n)$ is sharp. Moreover, suppose that $F(n)$ is a recursive upper bound of $H(n)$. Then there is no recursive function $\mathcal{D}(n)$ ordering to each n a finite set of natural numbers (represented for example as a string) larger than n such that for each n there is an x in $\mathcal{D}(n)$ with $F(x) \leq H(x)$. Notice that the function $\log n$ (or one almost equal to it) has this property for $K(n)$.

Solution. Suppose that such a function exists. Then we can select a sequence $n_1 < n_2 < \ldots$ of integers with the property that the sets $\mathcal{D}(n_i)$ are all disjoint. Let

$$a(n) = \sum_{x \in \mathcal{D}(n)} 2^{-F(x)}.$$

Then the sequence $a(n_k)$ is computable and $\sum_k a(n_k) < 1$. It is now easy to construct a computable subsequence $m_i = n_{k_i}$ of n_k and a computable sequence b_i such that $b_i/a(m_i) \to \infty$ and $\sum_i b_i < 1$. Let us define the semimeasure μ setting

$$\mu(x) = 2^{-F(x)} b_i / a(m_i)$$

for any x in $\mathcal{D}(m_i)$ and 0 otherwise. Then for any x in $\mathcal{D}(m_i)$ we have $H(x) \stackrel{+}{<} -\log \mu(x) = F(x) - \log c_i$ where $c_i = b_i/a(m_i) \to \infty$, so we arrived a contradiction with the assumption. □

Exercise 16 Prove that there is a recursive upper bound $F(n)$ of $H(n)$ and a constant c with the property that there are infinitely many natural numbers n such that for all $k > 0$, the quantity of numbers $x \leq n$ with $H(x) < F(x) - k$ is less than $cn2^{-k}$.

Solution. Use the upper bound G found in Theorem 1.7.4 and define $F(n) = \log n + G(\lfloor \log n \rfloor)$. The property follows from the facts that $\log n + H(\lfloor \log n \rfloor)$ is a sharp upper bound for $H(x)$ for "most" x less than n and that $G(k)$ is infinitely often close to $H(k)$. □

Exercise 17 Give an example of a computable sequence $a_n > 0$ of with the property that $\sum_n a_n < \infty$ but for any other computable sequence $b_n > 0$, if $b_n/a_n \to infty$ then $\sum_n b_n = \infty$.

Hint: Let r_n be a recursive, increasing sequence of rational numbers with $\lim_n r_n = \sum_x \mathbf{m}(x)$ and let $a_n = r_{n+1} - r_n$. □

Exercise 18 (Universal coding, Elias) Let $f(n) = \log_2 n + 2 \log_2 \log_2 n$. Show that when P runs over all nonincreasing probability distributions over N then

$$\lim_{H(P) \to \infty} H(P)^{-1} \sum_n P(n) f(n) = 1.$$

Exercise 19 (T. Cover) Let $\log_2^* n = \log_2 n + \log_2 \log_2 n + \ldots$ (all positive terms). Prove that

$$\sum_n 2^{-\log_2^* n} < \infty,$$

hence $H(n) \stackrel{+}{<} \log_2^* n$. For which logarithm bases does $H(n) \stackrel{+}{<} \log_2^* n$ hold?

Exercise 20 Prove that Kamae's result in Exercise 10 does not hold for $H(x \mid y)$.

Exercise 21 Prove that for each ε there is an m such that if $\mathcal{H}(P) > m$ then $|\sum_x P(x)H(x)/\mathcal{H}(P) - 1| < \varepsilon$.

Exercise 22 If a finite sequence x of length n has large complexity then its bits are certainly not predictable. Let $h(k, n) = -(k/n)log(k/n)-(1-(k/n))log(1-(k/n))$. A quantitative relation of this sort is the following. If, for some $k > n/2$, a program of length m can predict x_i from $x_1, \ldots, x_{i-1}$ for at least $n - k$ values of i then $K(x) < m + nh(k, n) + o(n)$.

Exercise 23 This is a more open-ended problem, having to do with the converse of the previous exercise. Show (if true) that for each c there is a d such that predictability of x at significantly more than $n/2 + c$ places is equivalent to the possibility to enclose x into a set of complexity $o(n)$ and size $n - dn$.

A Background from mathematics

A.1 Topology

In this section, we summarize the notions and results of topology that are needed in the main text.

A.1.1 Topological spaces

A topological space is a set of points with some kind of—not quantitatively expressed—closeness relation among them.

Definition A.1.1 A *topology* on a set X is defined by a class τ of its subsets called *open sets*. It is required that the empty set and X are open, and that arbitrary union and finite intersection of open sets is open. The pair (X, τ) is called a *topological space*. A set is called *closed* if its complement is open. ⌟

Having a set of open and closed sets allows us to speak about closure operations.

Definition A.1.2 A set B is called the *neighborhood* of a set A if B contains an open set that contains A. We denote by $\overline{A}, A^o$ the closure (the intersection of all closed sets containing A) and the interior of A (the union of all open sets in A) respectively. Let

$$\partial A = \overline{A} \setminus A^o$$

denote the boundary of set A. ⌟

An alternative way of defining a topological space is via a basis.

Definition A.1.3 A *basis* of a topological space is a subset β of τ such that every open set is the union of some elements of β. A *neighborhood* of a point is a basis element containing it. A *basis of neighborhoods of a point* x is a set N of neighborhoods of x with the property that each neighborhood of x contains an element of N. A *subbasis* is a subset σ of τ such that every open set is the union of finite intersections from σ. ⌟

Examples A.1.4

1. Let X be a set, and let β be the set of all points of X. The topology with basis β is the *discrete topology* of the set X. In this topology, every subset of X is open (and closed).
2. Let X be the real line $\mathbb{R}$, and let $\beta_{\mathbb{R}}$ be the set of all open intervals (a, b). The topology $\tau_{\mathbb{R}}$ obtained from this basis is the usual topology of the real line. When we refer to $\mathbb{R}$ as a topological space without qualification, this is the topology we will always have in mind.
3. Let $X = \overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$, and let $\beta_{\overline{\mathbb{R}}}$ consist of all open intervals (a, b) and in addition of all intervals of the forms $[-\infty, a)$ and $(a, \infty]$. It is clear how the space $\overline{\mathbb{R}}_+$ is defined.
4. Let X be the real line $\mathbb{R}$. Let $\beta^>_{\mathbb{R}}$ be the set of all open intervals $(-\infty, b)$. The topology with basis $\beta^>_{\mathbb{R}}$ is also a topology of the real line, different from the usual one. Similarly, let $\beta^<_{\mathbb{R}}$ be the set of all open intervals (b, ∞).
5. Let Σ be a finite or countable alphabet. On the space $\Sigma^{\mathbb{N}}$ of infinite sequences with elements in Σ, let $\tau_C = \{A\Sigma^{\mathbb{N}} : A \subseteq \Sigma^*\}$ be called the topology of the *Cantor space* (over Σ). Note that if a set E has the form $E = A\Sigma^{\mathbb{N}}$ where A is finite then E is both open and closed.

⌟

Starting from open sets, we can define some other kinds of set that are still relatively simple:

Definition A.1.5 A set is called a G_δ set if it is a countable intersection of open sets, and it is an F_σ set if it is a countable union of closed sets. ⌟

Different topologies over the same space have a natural partial order relation among them:

Definition A.1.6 A topology τ' on X is called *larger*, or *finer* than τ if $\tau' \supseteq \tau$. For two topologies τ_1, τ_2 over the same set X, we define the topology $\tau_1 \vee \tau_2 = \tau_1 \cap \tau_2$, and $\tau_1 \wedge \tau_2$ as the smallest topology containing $\tau_1 \cup \tau_2$. In the example topologies of the real numbers above, we have $\tau_{\mathbb{R}} = \tau^<_{\mathbb{R}} \wedge \tau^>_{\mathbb{R}}$. ⌟

Most topologies used in practice have some separion property.

Definition A.1.7 A topology is said to have the T_0 *property* if every point is determined by the class of open sets containing it. This is the weakest one of a number of other possible separation ⌟

Both our above example topologies of the real line have this property. All topologies considered in this survey will have the T_0 property. A stronger such property is the following:

Definition A.1.8 A space is called a *Hausdorff* space, or T_2 space, if for every pair of different points x, y there is a pair of disjoint open sets A, B with $x \in A$, $y \in B$. ⌟

The real line with topology $\tau^{>}_{\mathbb{R}}$ in Example A.1.4.4 above is not a Hausdorff space. A space is Hausdorff if and only if every open set is the union of closures of basis elements.

A.1.2 Continuity

We introduced topology in order to be able to speak about continuity:

Definition A.1.9 Given two topological spaces (X_i, τ_i) $(i = 1, 2)$, a function $f : X_1 \to X_2$ is called *continuous* if for every open set $G \subseteq X_2$ its inverse image $f^{-1}(G)$ is also open. If the topologies τ_1, τ_2 are not clear from the context then we will call the function (τ_1, τ_2)-continuous. Clearly, the property remains the same if we require it only for all elements G of a subbasis of X_2.

If there are two continuous functions between X and Y that are inverses of each other then the two spaces are called *homeomorphic*.

We say that function f is continuous *at point* x if for every neighborhood V of $f(x)$ there is a neighborhood U of x with $f(U) \subseteq V$. ⌟

Clearly, function f is continuous if and only if it is continuous in each point.

There is a natural sense in which every subset of a topological space is also a topological space:

Definition A.1.10 A *subspace* of a topological space (X, τ) is defined by a subset $Y \subseteq X$, and the topology $\tau_Y = \{G \cap Y : G \in \tau\}$, called the *induced* topology on Y. This is the smallest topology on Y making the identity mapping $x \mapsto x$ continuous. A partial function $f : X \to Z$ with $\mathrm{dom}(f) = Y$ is continuous iff $f : Y \to Z$ is continuous. ⌟

Given some topological spaces, we can also form larger ones using for example the product operation:

Definition A.1.11 For two topological spaces (X_i, τ_i) $(i = 1, 2)$, we define the *product topology* on their product $X \times Y$: this is the topology defined by the subbasis consisting of all sets $G_1 \times X_2$ and all sets $X_1 \times G_2$ with $G_i \in \tau_i$. The product topology is the smallest topology making the projection functions $(x, y) \mapsto x$, $(x, y) \mapsto y$ continuous. Given topological spaces X, Y, Z we call a two-argument function $f : X \times Y \mapsto Z$ continuous if it is continuous as a function from $X \times Y$ to Z. The product topology is defined similarly for the product $\prod_{i \in I} X_i$ of an arbitrary number of spaces, indexed by some index set I. We say that a function is $(\tau_1, \ldots, \tau_n, \mu)$-continuous if it is $(\tau_1 \times \cdots \times \tau_n, \mu)$-continuous. ⌟

Examples A.1.12

1. The space $\mathbb{R} \times \mathbb{R}$ with the product topology has the usual topology of the Euclidean plane.
2. Let X be a set with the discrete topology, and $X^{\mathbb{N}}$ the set of infinite sequences with elements from X, with the product topology. A basis of this topology is provided by all sets of the form $uX^{\mathbb{N}}$ where $u \in X^*$. The elements of this basis are closed as well as open. When $X = \{0, 1\}$ then this topology is the usual topology of infinite binary sequences.

⌟

In some special cases, one of the topologies in the definition of continuity is fixed and known in advance:

Definition A.1.13 A real function $f : X_1 \to \mathbb{R}$ is called continuous if it is $(\tau_1, \tau_{\mathbb{R}})$-continuous. ⌟

A.1.3 Semicontinuity

For real functions, a restricted notion of continuity is often useful.

Definition A.1.14 A function $f : X \to \overline{\mathbb{R}}$ is *lower semicontinuous* if the set $\{(x, r) \in X \times \overline{\mathbb{R}} : f(x) > r\}$ is open. The definition of upper semicontinuity is similar. Lower semicontinuity on subspaces is defined similarly to continuity on subspaces. ⌟

Clearly, a real function f is continuous if and only if it is both lower and upper semicontinuous. The requirement of lower semicontinuity of f is equivalent to saying that for each $r \in \mathbb{R}$, the set $\{x : f(x) > r\}$ is open. This can be seen to be equivalent to the following characterization.

Proposition A.1.15 *A real function f over $\mathbf{X} = (X, \tau)$ is lower semicontinuous if and only if it is $(\tau, \tau_{\mathbb{R}}^{<})$-continuous.*

Example A.1.16 The indicator function $1_G(x)$ of an arbitrary open set G is lower semicontinuous. ⌟

The following is easy to see.

Proposition A.1.17 *The supremum of any set of lower semicontinuous functions is lower semicontinuous.*

The following representation is then also easy to see.

Proposition A.1.18 *Let X be a topological space with basis β. The function $f : X \to \overline{\mathbb{R}}_+$ is lower semicontinuous if and only if there is a function $g : \beta \to \overline{\mathbb{R}}_+$ with $f(x) = \sup_{x \in \beta} g(\beta)$.*

Corollary A.1.19 *Let X be a topological space with basis β and $f : X \to \overline{\mathbb{R}}_+$ a lower semicontinuous function defined on a subset D of X. Then f can be extended in a lower semicontinuous way to the whole space X.*

Proof. Indeed by the above proposition there is a function $g : \beta \to \overline{\mathbb{R}}_+$ with $f(x) = \sup_{x \in \beta} g(\beta)$ for all $x \in D$. Let us define f by this same formula for all $x \in X$. □

In the important special case of Cantor spaces, the basis is given by the set of finite sequences. In this case we can also require the function $g(w)$ to be monotonic in the words w:

Proposition A.1.20 *Let $X = \Sigma^{\mathbb{N}}$ be a Cantor space as defined in Example A.1.4.5. Then $f : X \to \overline{\mathbb{R}}_+$ is lower semicontinuous if and only if there is a function $g : \Sigma^* \to \overline{\mathbb{R}}_+$ monotonic with respect to the relation $u \sqsubseteq v$, with $f(\xi) = \sup_{u \sqsubseteq \xi} g(u)$.*

A.1.4 Compactness

There is an important property of topological spaces that, when satisfied, has many useful implications.

Definition A.1.21 Let (X, τ) be a topological space, and B a subset of X. An *open cover* of B is a family of open sets whose union contains B. A subset K of X is said to be *compact* if every open cover of K has a finite subcover. ⌟

Compact sets have many important properties: for example, a continuous function over a compact set is bounded.

Example A.1.22

1. Every finite discrete space is compact. An infinite discrete space $\mathbf{X} = (X, \tau)$ is not compact, but it can be turned into a compact space $\overline{\mathbf{X}}$ by adding a new element called ∞: let $\overline{X} = X \cup \{\infty\}$, and $\overline{\tau} = \tau \cup \{\overline{X} \setminus A : A \subset X \text{ closed }\}$. More generally, this simple operation can be performed with every space that is *locally compact*, that each of its points has a compact neighborhood.
2. In a finite-dimensional Euclidean space, every bounded closed set is compact.
3. It is known that if $(\mathbf{X}_i)_{i \in I}$ is a family of compact spaces then their direct product is also compact.

⌟

There are some properties that are equivalent to compactness in simple cases, but not always:

Definition A.1.23 A subset K of X is said to be *sequentially compact* if every sequence in K has a convergent subsequence with limit in K. The space is *locally compact* if every point has a compact neighborhood. ⌟

A.1.5 Metric spaces

Metric spaces are topological spaces with more structure: in them, the closeness concept is quantifiable. In our examples for metric spaces, and later in our treatment of the space of probability measures, we refer to [5].

Definition A.1.24 A *metric space* is given by a set X and a distance function $d : X \times X \to \mathbb{R}_+$ satisfying the *triangle inequality* $d(x, z) \leq d(x, y) + d(y, z)$ and also property that $d(x, y) = 0$ implies $x = y$. For $r \in \mathbb{R}_+$, the sets

$$B(x, r) = \{y : d(x, y) < r\}, \quad \overline{B}(x, r) = \{y : d(x, y) \leq r\}$$

are called the open and closed *balls* with radius r and center x.

A metric space is *bounded* when $d(x, y)$ has an upper bound on X. ⌟

A metric space is also a topological space, with the basis that is the set of all open balls. Over this space, the distance function $d(x, y)$ is obviously continuous.

Each metric space is a Hausdorff space; moreover, it has the following stronger property.

Definition A.1.25 A topological space is said to have the T_3 *property* if for every pair of different points x, y there is a continuous function $f : X \to \mathbb{R}$ with $f(x) \neq f(y)$. ⌟

To see that metric spaces are T_3, take $f(z) = d(x, z)$.

Definition A.1.26 A topological space is called *metrizable* if its topology can be derived from some metric space. ⌟

It is known that a topological space is metrizable if and only if it has the T_3 property.

Notation A.1.27 For an arbitrary set A and point x let

$$d(x, A) = \inf_{y \in A} d(x, y),$$
$$A^\varepsilon = \{x : d(x, A) < \varepsilon\}. \tag{A.1.1}$$

⌟

Examples A.1.28

1. A discrete topological space X can be turned into a metric space as follows: $d(x, y) = 0$ if $x = y$ and 1 otherwise.

2. The real line with the distance $d(x, y) = |x - y|$ is a metric space. The topology of this space is the usual topology $\tau_{\mathbb{R}}$ of the real line.
3. The space $\mathbb{R} \times \mathbb{R}$ with the Euclidean distance gives the same topology as the product topology of $\mathbb{R} \times \mathbb{R}$.
4. An arbitrary set X with the distance $d(x, y) = 1$ for all pairs x, y of different elements, is a metric space that induces the discrete topology on X.
5. Let X be a bounded metric space, and let $Y = X^{\mathbb{N}}$ be the set of infinite sequences $x = (x_1, x_2, \dots)$ with distance function $d^{\mathbb{N}}(x, y) = \sum_i 2^{-i} d(x_i, y_i)$. The topology of this space is the same as the product topology defined in Example A.1.12.2.
6. Specializing the above example, if Σ is the discrete space defined in Example 1 above then we obtain a metrization of the Cantor space of Example A.1.45. For every finite sequence $x \in \Sigma^*$ and every infinite sequence $\xi \sqsupseteq x$ the ball $B(\xi, 2^{-l(x)})$ is equal to a basis element that is the open-closed cylinder set $x\Sigma^{\mathbb{N}}$.
7. Let X be a metric space, and let $Y = X^{\mathbb{N}}$ be the set of infinite bounded sequences $x = (x_1, x_2, \dots)$ with distance function $d(x, y) = \sup_i d(x_i, y_i)$.
8. Let X be a topological space, and let $C(X)$ be the set of bounded continuous functions over X with distance function $d'(f, g) = \sup_x d(f(x), g(x))$. A special case is $C[0, 1]$ where the interval $[0, 1]$ of real numbers has the usual topology.
9. Let l_2 be the set of infinite sequences $x = (x_1, x_2, \dots)$ of real numbers with the property that $\sum_i x_i^2 < \infty$. The metric is given by the distance $d(x, y) = (\sum_i |x_i - y_i|^2)^{1/2}$.

⌟

In metric spaces, certain previously defined topological objects have richer properties.

Examples A.1.29 Each of the following facts holds in metric spaces and is relatively easy to prove.

1. Every closed set is a G_δ set (and every open set is an F_σ set).
2. A set is compact if and only if it is sequentially compact.
3. A set is compact if and only if it is closed and for every ε, there is a finite set of ε-balls (balls of radius ε) covering it.

⌟

In metric spaces, the notion of continuity can be strengthened.

Definition A.1.30 A function $f : X \to Y$ between metric spaces X, Y is *uniformly continuous* if for each $\varepsilon > 0$ there is a $\delta > 0$ such that $d_X(a, b) < \delta$ implies $d_Y(f(a), f(b)) < \varepsilon$. ⌟

It is known that over a compact metric space, every continuous function is uniformly continuous.

Definition A.1.31 (Lipschitz) Given a function $f : X \to Y$ between metric spaces and $\beta > 0$, let $\mathrm{Lip}_\beta(X, Y)$ denote the set of functions (called the Lipschitz(β) functions, or simply Lipschitz functions) satisfying

$$d_Y(f(x), f(y)) \leq \beta d_X(x, y). \tag{A.1.2}$$

Let $\mathrm{Lip}(X) = \mathrm{Lip}(X, \mathbb{R}) = \bigcup_\beta \mathrm{Lip}_\beta$ be the set of real Lipschitz functions over X. ⌟

As an example, every differentiable real function $f(x)$ with $|f'(x)| \leq 1$ everywhere is a Lipschitz(1) function.

All these functions are uniformly continuous.

We introduce a certain fixed, enumerated sequence of Lipschitz functions that will be used frequently as "building blocks" of other functions.

Definition A.1.32 (Hat functions) Let

$$g_{u,r,\varepsilon}(x) = |1 - |d(x, u) - r|^+/\varepsilon|^+.$$

This is a continuous function that is 1 in the ball $B(u, r)$, it is 0 outside the ball $B(u, r+\varepsilon)$, and takes intermediate values in between. It is clearly a Lipschitz($1/\varepsilon$) function.

If a dense set D is fixed, let $\mathcal{F}_0 = \mathcal{F}_0(D)$ be the set of functions of the form $g_{u,r,1/n}$ where $u \in D$, r is rational, $n = 1, 2, \ldots$. Let $\mathcal{F}_1 = \mathcal{F}_1(D)$ be the maximum of a finite number of elements of $\mathcal{F}_0(D)$. Each element f of $\mathcal{F}_1$ is bounded between 0 and 1. Let

$$\mathcal{E} = \mathcal{E}(D) = \{g_1, g_2, \ldots\} \supset \mathcal{F}_1 \tag{A.1.3}$$

be the smallest set of functions containing $\mathcal{F}_0$ and the constant 1, and closed under $\vee$, $\wedge$ and rational linear combinations. For each element of $\mathcal{E}$, from its definition we can compute a bound β such that $f \in \mathrm{Lip}_\beta$. ⌟

For the effective representation of points in a topological space the following properties are important.

Definition A.1.33 A topological space has the *first countability property* if each point has a countable basis of neighborhoods. ⌟

Every metric space has the first countability property since we can restrict ourselves to balls with rational radius.

Definition A.1.34 Given a topological space (X, τ) and a sequence $x = (x_1, x_2, \ldots)$ of elements of X, we say that x *converges* to a point y if for every

neighborhood G of y there is a k such that for all $m > k$ we have $x_m \in G$. We will write $y = \lim_{n\to\infty} x_n$. ⌟

It is easy to show that if spaces (X_i, τ_i) $(i = 1, 2)$ have the first countability property then a function $f : X \to Y$ is continuous if and only if for every convergent sequence (x_n) we have $f(\lim_n x_n) = \lim_n f(x_n)$.

Definition A.1.35 A topological space has the *second countability property* if the whole space has a countable basis. ⌟

For example, the space $\mathbb{R}$ has the second countability property for all three topologies $\tau_{\mathbb{R}}$, $\tau_{\mathbb{R}}^{<}$, $\tau_{\mathbb{R}}^{>}$. Indeed, we also get a basis if instead of taking all intervals, we only take intervals with rational endpoints. On the other hand, the metric space of Example A.1.28.7 does not have the second countability property.

Definition A.1.36 In a topological space (X, τ), a set B of points is called *dense* at a point x if it intersects every neighborhood of x. It is called *everywhere dense*, or *dense*, if it is dense at every point. A metric space is called *separable* if it has a countable everywhere dense subset. ⌟

It is easy to see that a metric space is separable if and only if as a topological space it has the second countability property.

Example A.1.37 In Example A.1.28.8, for $X = [0, 1]$, we can choose as our everywhere dense set the set of all polynomials with rational coefficients, or alternatively, the set of all piecewise linear functions whose graph has finitely many nodes at rational points.

More generally, let X be a *compact separable* metric space with a dense set D. Then it can be shown that in the metric space $C(X)$, the set of functions $\mathcal{E}(D)$ introduced in Definition A.1.32 is dense, and turns it into a complete (not necessarily compact!) separable metric space. ⌟

Definition A.1.38 In a metric space, let us call a sequence $x_1, x_2, \ldots$ a *Cauchy* sequence if for all $i < j$ we have $d(x_i, x_j) < 2^{-i}$. ⌟

It is easy to see that if an everywhere dense set D is given then every element of the space can be represented as the limit of a Cauchy sequence of elements of D. But not every Cauchy sequence needs to have a limit.

Definition A.1.39 A metric space is called *complete* if every Cauchy sequence in it has a limit. ⌟

For example, if X is the real line with the point 0 removed then X is not complete, since there are Cauchy sequences converging to 0, but 0 is not in X.

It is well-known that every metric space can be embedded (as an everywhere dense subspace) into a complete space.

Example A.1.40 Consider the set $D[0, 1]$ of functions over $[0, 1]$ that are right continuous and have left limits everywhere. The book [5] introduces two different metrics for them: the Skorohod metric d and the d_0 metric. In both metrics, two functions are close if a slight monotonic continuous deformation of the coordinates makes them uniformly close. But in the d_0 metric, the slope of the deformation must be close to 1. It is shown that the two metrics give rise to the same topology; however, the space with metric d is not complete, and the space with metric d_0 is. ⌟

We will develop the theory of randomness over separable complete metric spaces. This is a wide class of spaces encompassing most spaces of practical interest. The theory would be simpler if we restricted it to compact or locally compact spaces; however, some important spaces like $C[0, 1]$ (the set of continuouos functions over the interval $[0, 1]$, with the maximum difference as their distance) are not locally compact.

A.2 Measures

For a survey of measure theory, see for example [39].

A.2.1 Set algebras

Event in a probability space are members of a class of sets that is required to be a so-called σ-algebra (sigma-algebra).

Definition A.2.1 A (Boolean set-) *algebra* is a set of subsets of some set X closed under intersection and complement (and then, of course, under union). It is a *σ-algebra* (sigma-algebra) if it is also closed under countable intersection (and then, of course, under countable union). A *semialgebra* is a set $\mathcal{L}$ of subsets of some set X closed under intersection, with the property that the complement of every element of $\mathcal{L}$ is the disjoint union of a finite number of elements of $\mathcal{L}$. ⌟

If $\mathcal{L}$ is a semialgebra then the set of finite unions of elements of $\mathcal{L}$ is an algebra.

Examples A.2.2

1. The set $\mathcal{L}_1$ of left-closed intervals of the line (including intervals of the form $(-\infty, a)$) is a semialgebra.
2. The set $\mathcal{L}_2$ of all intervals of the line (which can be open, closed, left-closed or right-closed), is a semialgebra.
3. In the set $\{0, 1\}^{\mathbb{N}}$ of infinite 0-1-sequences, the set $\mathcal{L}_3$ of all subsets of the form $u\{0, 1\}^{\mathbb{N}}$ with $u \in \{0, 1\}^*$, is a semialgebra.

4. The σ-algebra $\mathcal{B}$ generated by $\mathcal{L}_1$, is the same as the one generated by $\mathcal{L}_2$, and is also the same as the one generated by the set of all open sets: it is called the family of *Borel sets* of the line. The Borel sets of the extended real line $\overline{\mathbb{R}}$ are defined similarly.
5. More generally, the class of Borel sets in an arbitrary topological space is the smallest σ-algebra containing all open sets.
6. Given σ-algebras $\mathcal{A}, \mathcal{B}$ in sets X, Y, the product σ-algebra $\mathcal{A} \times \mathcal{B}$ in the space $X \times Y$ is the one generated by all elements $A \times Y$ and $X \times B$ for $A \in \mathcal{A}$ and $B \in \mathcal{B}$.

⌟

A.2.2 Measures

Probability is an example of the more general notion of a measure.

Definition A.2.3 A *measurable space* is a pair $(X, \mathcal{S})$ where $\mathcal{S}$ is a σ-algebra of sets of X. A *measure* on a measurable space $(X, \mathcal{S})$ is a function $\mu : B \to \overline{\mathbb{R}}_+$ that is σ-*additive*: this means that for every countable family $A_1, A_2, \ldots$ of disjoint elements of $\mathcal{S}$ we have $\mu(\bigcup_i A_i) = \sum_i \mu(A_i)$. A measure μ is σ-*finite* if the whole space is the union of a countable set of subsets whose measure is finite. It is *finite* if $\mu(X) < \infty$. It is a *probability measure* if $\mu(X) = 1$. ⌟

Example A.2.4 (Delta function) For any point x, the measure δ_x is defined as follows:

$$\delta_x(A) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{otherwise.} \end{cases}$$

⌟

Definition A.2.5 If μ is a measure, a point x is called an *atom* if $\mu(x) > 0$. ⌟

Generally, we will consider measures over either a countable set (a discrete measure for which the union of atoms has total measure) or an uncountable one, with no atoms. But mixed cases are possible.

It is important to understand how a measure can be defined in practice. Algebras are generally simpler to grasp constructively than σ-algebras; semialgebras are yet simpler. Suppose that μ is defined over a semialgebra $\mathcal{L}$ and is additive. Then it can always be uniquely extended to an additive function over the algebra generated by $\mathcal{L}$. The following is an important theorem of measure theory.

Proposition A.2.6 (Caratheodory's extension theorem) *Suppose that a nonnegative set function defined over a semialgebra $\mathcal{L}$ is σ-additive. Then it can be extended uniquely to the σ-algebra generated by $\mathcal{L}$.*

Examples A.2.7

1. Let x be point and let $\mu(A) = 1$ if $x \in A$ and 0 otherwise. In this case, we say that μ is *concentrated* on the point x.
2. Consider the the line $\mathbb{R}$, and the algebra $\mathcal{L}_1$ defined in Example A.2.2.1. Let $f : \mathbb{R} \to \mathbb{R}$ be a monotonic real function. We define a set function over $\mathcal{L}_1$ as follows. Let $[a_i, b_i)$, $(i = 1, \ldots, n)$ be a set of disjoint left-closed intervals. Then $\mu(\bigcup_i [a_i, b_i)) = \sum_i f(b_i) - f(a_i)$. It is easy to see that μ is additive. It is σ-additive if and only if f is left-continuous.
3. Let $B = \{0, 1\}$, and consider the set $B^{\mathbb{N}}$ of infinite 0-1-sequences, and the semialgebra $\mathcal{L}_3$ of Example A.2.2.3. Let $\mu : B^* \to \mathbb{R}_+$ be a function. Let us write $\mu(uB^{\mathbb{N}}) = \mu(u)$ for all $u \in B^*$. Then it can be shown that the following conditions are equivalent: μ is σ-additive over $\mathcal{L}_3$; it is additive over $\mathcal{L}_3$; the equation $\mu(u) = \mu(u0) + \mu(u1)$ holds for all $u \in B^*$.
4. The nonnegative linear combination of any finite number of measures is also a measure. In this way, it is easy to construct arbitrary measures concentrated on a finite number of points.
5. Given two measure spaces $(X, \mathcal{A}, \mu)$ and $(Y, \mathcal{B}, \nu)$ it is possible to define the product measure $\mu \times \nu$ over the measureable space $(X \times Y, \mathcal{A} \times \mathcal{B})$. The definition is required to satisfy $\mu \times \nu(A \times B) = \mu(A) \times \nu(B)$, and is determined uniquely by this condition. If ν is a probability measure then, of course, $\mu(A) = \mu \times \nu(A \times Y)$.

⌟

Let us finally define measureable functions.

Definition A.2.8 (Measureable functions) Let $(X, \mathcal{A})$, $(Y, \mathcal{B})$ be two measureable spaces. A function $f : X \to Y$ is called *measureable* if and only if $f^{-1}(E) \in \mathcal{A}$ for all $E \in \mathcal{B}$. ⌟

The following is easy to check.

Proposition A.2.9 *Let $(X, \mathcal{A})$ be a measureable space and $(\mathbb{R}, \mathcal{B})$ be the measureable space of the real numbers, with the Borel sets. Then $f : X \to \mathbb{R}$ is measureable if and only if all sets of the form $f^{-1}((r, \infty)) = \{x : f(x) > r\}$ are measureable, where r is a rational number.*

Remark A.2.10 Example A.2.7.3 shows a particularly attractive way to define measures. Keep splitting the values $\mu(u)$ in an arbitrary way into $\mu(u0)$ and $\mu(u1)$, and the resulting values on the semialgebra define a measure. Example A.2.7.2 is less attractive, since in the process of defining μ on all intervals and only keeping track of finite additivity, we may end up with a monotonic function that is not left

continuous, and thus with a measure that is not σ-additive. In the subsection on probability measures over a metric space, we will find that even on the real line, there is a way to define measures in a step-by-step manner, and only checking for consistency along the way. ⌟

A probability space, according to the axioms introduced by Kolmogorov, is just a measureable space with a normed measure.

Definition A.2.11 A *probability space* is a triple $(X, \mathcal{S}, P)$ where $(X, \mathcal{S})$ is a measurable space and P is a probability measure over it.

Let $(X_i, \mathcal{S}_i)$ $(i = 1, 2)$ be measurable spaces, and let $f : X \to Y$ be a mapping. Then f is *measurable* if and only if for each element B of $\mathcal{S}_2$, its inverse image $f^{-1}(B)$ is in $\mathcal{S}_1$. If μ_1 is a measure over $(X_1, \mathcal{S}_1)$ then μ_2 defined by $\mu_2(A) = \mu_1(f^{-1}(A))$ is a measure over X_2 called the measure *induced* by f. ⌟

A.2.3 Integral

The notion of integral also generalizes to arbitrary measures, and is sometimes also used to define measures.

First we define integral on very simple functions.

Definition A.2.12 A measurable function $f : X \to \mathbb{R}$ is called a *step function* if its range is finite.

The set of step functions is closed with respect to linear combinations and also with respect to the operations $\wedge$, $\vee$. Any such set of functions is called a *Riesz space*. ⌟

Definition A.2.13 Given a step function f which takes values x_i on sets A_i, and a finite measure μ, we define

$$\mu(f) = \mu f = \int f \, d\mu = \int f(x)\mu(dx) = \sum_i x_i \, \mu(A_i).$$

⌟

This is a linear positive functional on the set of step functions. Moreover, it can be shown that it is continuous on monotonic sequences: if $f_i \searrow 0$ then $\mu f_i \searrow 0$. The converse can also be shown: Let μ be a linear positive functional on step functions that is continuous on monotonic sequences. Then the set function $\mu(A) = \mu(1_A)$ is a finite measure.

Proposition A.2.14 *Let $\mathcal{E}$ be any Riesz space of functions with the property that $1 \in \mathcal{E}$. Let μ be a positive linear functional on $\mathcal{E}$ continuous on monotonic sequences, with $\mu 1 = 1$. The functional μ can be extended to the set $\mathcal{E}_+$ of monotonic limits of nonnegative*

elements of $\mathcal{E}$, by continuity. In case when $\mathcal{E}$ is the set of all step functions, the set $\mathcal{E}_+$ is the set of all nonnegative measurable functions.

Now we extend the notion of integral to a wider class of functions.

Definition A.2.15 Let us fix a finite measure μ over a measurable space $(X, \mathcal{S})$. A measurable function f is called *integrable* with respect to μ if $\mu|f|^+ < \infty$ and $\mu|f|^- < \infty$. In this case, we define $\mu f = \mu|f|^+ - \mu|f|^-$. ⌟

The set of integrable functions is a Riesz space, and the positive linear functional μ on it is continuous with respect to monotonic sequences. The continuity over monotonic sequences also implies the following theorem.

Proposition A.2.16 (Bounded convergence theorem) *Suppose that functions f_n are integrable and $|f_n| < g$ for some integrable function g. Then $f = \lim_n f_n$ is integrable and $\mu f = \lim_n \mu f_n$.*

Definition A.2.17 Two measurable functions f, g are called *equivalent* with respect to measure μ if $\mu(f - g) = 0$. ⌟

For two-dimensional integration, the following theorem holds.

Proposition A.2.18 (Fubini theorem) *Suppose that function $f(\cdot, \cdot)$ is integrable over the space $(X \times Y, \mathcal{A} \times \mathcal{B}, \mu \times \nu)$. Then for μ-almost all x, the function $f(x, \cdot)$ is integrable over $(Y, \mathcal{B}, \nu)$, and the function $x \mapsto \nu^y f(x, y)$ is integrable over $(X, \mathcal{A}, \mu)$ with $(\mu \times \nu)f = \mu^x \mu^y f$.*

To express a continuity property of measures, we can say the following (recall the definition of $C(X)$ in Example A.1.28.8).

Proposition A.2.19 *Let X be a metric space and μ a measure. Then μ is a bounded (and thus continuous) linear functional over the space $C(X)$.*

A.2.4 Density

When does one measure have a density function with respect to another?

Definition A.2.20 Let μ, ν be two measures over the same measurable space. We say that ν is *absolutely continuous* with respect to μ, or that μ *dominates* ν, if for each set A, $\mu(A) = 0$ implies $\nu(A) = 0$. ⌟

Every nonnegative integrable function f defines a new measure ν via the formula $\nu(A) = \mu(f \cdot 1_A)$. This measure ν is absolutely continuous with respect to μ. The Radon-Nikodym theorem says that the converse is also true.

Proposition A.2.21 (Radon-Nikodym theorem) *If ν is dominated by μ then there is a nonnegative integrable function f such that $\nu(A) = \mu(f \cdot 1_A)$ for all measurable sets*

A. The function f is defined uniquely to within equivalence with respect to μ.

Definition A.2.22 The function f of the Radom-Nikodym Theorem above is called the *density* of ν with respect to μ. We will denote it by

$$f(x) = \frac{\mu(dx)}{\nu(dx)} = \frac{d\mu}{d\nu}.$$

⌟

The following theorem is also standard.

Proposition A.2.23 (Chain rule and inverse function)

1. *Let μ, ν, η be measures such that η is absolutely continuous with respect to μ and μ is absolutely continuous with respect to ν. Then the "chain rule" holds:*

$$\frac{d\eta}{d\nu} = \frac{d\eta}{d\mu}\frac{d\mu}{d\nu}. \tag{A.2.1}$$

2. *If $\frac{\nu(dx)}{\mu(dx)} > 0$ for all x then μ is also absolutely continuous with respect to ν and $\frac{\mu(dx)}{\nu(dx)} = \left(\frac{\nu(dx)}{\mu(dx)}\right)^{-1}$.*

There is a natural distance to be used between measures, though later we will see that it is not the preferred one in metric spaces.

Definition A.2.24 (Total variation distance) Let μ, ν be two measures, then both are dominated by some measure η (for example by $\eta = \mu + \nu$). Let their densities with respect to η be f and g. Then we define the *total variation distance* of the two measures as

$$D(\mu, \nu) = \eta(|f - g|).$$

It is independent of the dominating measure η. ⌟

Example A.2.25 Suppose that the space X can be partitioned into disjoint sets A, B such that $\nu(A) = \mu(B) = 0$. Then $D(\mu, \nu) = \mu(A) + \nu(B) = \mu(X) + \nu(X)$. ⌟

A.2.5 Random transitions

What is just a transition matrix in case of a Markov chain also needs to be defined more carefully in the non-discrete cases. We follow the definition given in [39].

Definition A.2.26 Let $(X, \mathcal{A})$, $(Y, \mathcal{B})$ be measureable spaces (defined in Subsection A.2.2). Suppose that a family of probability measures $\Lambda = \{\lambda_x : x \in X\}$ on $\mathcal{B}$ is given. We call it a *probability kernel*, (or Markov kernel, or conditional distribution) if the map $x \mapsto \lambda_x B$ is measurable for each $B \in \mathcal{B}$. ⌟

When X, Y are finite sets then λ is a Markov transition matrix. The following theorem shows that λ assigns a joint distribution over the space $(X \times Y, \mathcal{A} \times \mathcal{B})$ to each input distribution μ.

Proposition A.2.27 *For each nonnegative $\mathcal{A} \times \mathcal{B}$-measureable function f over $X \times Y$,*

1. *The function $y \mapsto f(x, y)$ is $\mathcal{B}$-measurable for each fixed x.*
2. *The function $x \mapsto \lambda_x^y f(x, y)$ is $\mathcal{A}$-measurable.*
3. *The integral $f \mapsto \mu^x \lambda_x^y f(x, y)$ defines a measure on $\mathcal{A} \times \mathcal{B}$.*

The above proposition allows to define a mapping over measures.

Definition A.2.28 According to the above proposition, given a probability kernel Λ, to each measure μ over $\mathcal{A}$ corresponds a measure over $\mathcal{A} \times \mathcal{B}$. We will denote its marginal over $\mathcal{B}$ as

$$\Lambda^* \mu. \tag{A.2.2}$$

For every measurable function $g(y)$ over Y, we can define the measurable function $f(x) = \lambda_x g = \lambda_x^y g(y)$: we write

$$f = \Lambda g. \tag{A.2.3}$$

⌟

The operator Λ is linear, and monotone with $\Lambda 1 = 1$. By these definitions, we have

$$\mu(\Lambda g) = (\Lambda^* \mu) g. \tag{A.2.4}$$

An example is the simple case of a deterministic mapping:

Example A.2.29 Let $h : X \to Y$ be a measureable function, and let λ_x be the measure $\delta_{h(x)}$ concentrated on the point $h(x)$. This operator, denoted Λ_h is, in fact, a deterministic transition, and we have $\Lambda_h g = g \circ h$. In this case, we will simplify the notation as follows:

$$h^* \mu = \Lambda_h^*.$$

⌟

A.2.6 Probability measures over a metric space

We follow the exposition of [5]. Whenever we deal with probability measures on a metric space, we will assume that our metric space is complete and separable (Polish space).

Let $\mathbf{X} = (X, d)$ be a complete separable metric space. Then $\mathbf{X}$ gives rise to a measurable space, where the measurable sets are its Borel sets. It can be shown that, if A is a Borel set and μ is a finite measure then there are sets $F \subseteq A \subseteq G$ where F is an F_σ set, G is a G_δ set, and $\mu(F) = \mu(G)$.

It can be shown that a measure is determined by its values on the elements of any a basis of open sets closed under intersections. The following proposition follows then essentially from Proposition A.2.6.

Proposition A.2.30 *Let $\mathcal{B}$ be a basis of open sets closed under intersections. Let $\mathcal{B}^*$ be the set algebra generated by this basis and let μ be any σ-additive set function on $\mathcal{B}^*$ with $\mu(X) = 1$. Then μ can be extended uniquely to a probability measure.*

Weak topology

One can introduce the notion of convergence of measures in a number of ways. We have already introduced the total variation distance in Definition A.2.24 above. But in some cases, the requirement of being close in this distance is too strong. Let

$$\mathcal{M}(\mathbf{X})$$

be the set of probability measures on the metric space $\mathbf{X}$. Let x_n be a sequence of points converging to point x but with $x_n \neq x$. We would like to say that the delta measure δ_{x_n} (concentrated on point x_n, see Example A.2.4) converges to δ_x. But the total variation distance $D(\delta_{x_n}, \delta_x)$ is 2 for all n.

Definition A.2.31 (Weak convergence) We say that a sequence of probability measures μ_n over a metric space (X, d) *weakly converges* to measure μ if for all bounded continuous real functions f over X we have $\mu_n f \to \mu f$.

For a bounded continuous function f and real numbers c let

$$A_{f,c} = \{\mu : \mu f < c\}$$

⌟

A *topology of weak convergence* $(\mathcal{M}, \tau_w)$ can be defined using a number of different subbases. The one used in the original definition is the subbasis consisting of all sets of the form $A_{f,c}$ above.

We also get a subbasis (see for example [39]) if we restrict ourselves to the set $\mathrm{Lip}(X)$ of Lipschitz functions defined in (A.1.2). Another possible subbasis giving rise to the same topology consists of all sets of the form

$$B_{G,c} = \{\mu : \mu(G) > c\} \tag{A.2.5}$$

for open sets G and real numbers c. Let us find some countable subbases. Since the space $\mathbf{X}$ is separable, there is a sequence $U_1, U_2, \ldots$ of open sets that forms a basis of $\mathbf{X}$. Then we can restrict the subbasis of the space of measures to those sets $B_{G,c}$ where G is the union of a finite number of basis elements U_i of $\mathbf{X}$ and c is rational. This way, the space $(\mathcal{M}, \tau_w)$ itself has the second countability property.

It is more convenient to define a countable subbasis using bounded continuous functions f, since the function $\mu \mapsto \mu f$ is continuous on such functions, while $\mu \mapsto \mu U$ is typically not continuous when U is an open set.

Example A.2.32 If $\mathbf{X} = \mathbb{R}$ and U is the open interval $(0, 1)$, the sequence of probability measures $\delta_{1/n}$ (concentrated on $1/n$) converges to δ_0, but $\delta_{1/n}(U) = 1$, and $\delta_0(U) = 0$. ⌟

For some fixed dense set D, let $\mathcal{F}_1 = \mathrm{F}_1(D)$ be the set of functions introduced in Definition A.1.32.

Definition A.2.33 We say that a set A is a *continuity set* of measure μ if $\mu(\partial A) = 0$: the boundary of A has measure 0. ⌟

Proposition A.2.34 *The following conditions are equivalent:*

1. μ_n *weakly converges to* μ.
2. $\mu_n f \to \mu f$ *for all* $f \in \mathcal{F}_1$.
3. *For every Borel set* A*, that is a continuity set of* μ*, we have* $\mu_n(A) \to \mu(A)$.
4. *For every closed set* F*,* $\liminf_n \mu_n(F) \geq \mu(F)$.
5. *For every open set* G*,* $\limsup_n \mu_n(G) \leq \mu(G)$.

Definition A.2.35 To define the topological space $\mathcal{M}(X)$ of the set of measures over the metric space X, we choose as subbasis

$$\sigma_{\mathcal{M}} \tag{A.2.6}$$

the sets $\{\mu : \mu f < r\}$ and $\{\mu : \mu f > r\}$ for all $f \in \mathcal{F}_1$ and $r \in \mathbb{Q}$. ⌟

The simple functions we introduced can also be used to define measure and integral in themselves. Recall the definition of the set $\mathcal{E}$ in (A.1.3).T This set is a Riesz space as defined in Subsection A.2.3. A reasoning combining Propositions A.2.6 and A.2.14 gives the following.

Proposition A.2.36 *Suppose that a positive linear functional* μ *with* $\mu 1 = 1$ *is defined on* $\mathcal{E}$ *that is continuous with respect to monotone convergence. Then* μ *can be extended uniquely to a probability measure over* $\mathbf{X}$ *with* $\mu f = \int f(x)\mu(dx)$ *for all* $f \in \mathcal{E}$.

Having a topology over the set of measures we can also extend Proposition A.2.19:

Proposition A.2.37 *Let* X *be a complete separable metric space and* $\mathcal{M}(X)$ *the space of bounded measures over* X *with the weak topology. The function* $(\mu, f) \mapsto \mu f$ *is a continuous function* $\mathcal{M}(X) \times C(X) \to \mathbb{R}$.

As mentioned above, for an open set G the value $\mu(G)$ is not a continuous function of the measure μ. We can only say the following:

Proposition A.2.38 *Let X be a complete separable metric space, and $\mathcal{M}(X)$ the space of bounded measures over X with the weak topology, and $G \subseteq X$ an open set. The function $\mu \mapsto \mu(G)$ is lower semicontinuous.*

Distances for the weak topology

The definition of measures in the style of Proposition A.2.36 is not sufficiently constructive. Consider a gradual definition of the measure μ, extending it to more and more elements of $\mathcal{E}$, while keeping the positivity and linearity property. It can happen that the function μ we end up with in the limit, is not continuous with respect to monotone convergence. Let us therefore metrize the space of measures: then an arbitrary measure can be defined as the limit of a Cauchy sequence of simple meaures.

One metric that generates the topology of weak convergence is the following.

Definition A.2.39 (Prokhorov distance) The *Prokhorov distance* $\rho(\mu, \nu)$ of two measures is the infimum of all those ε for which, for all Borel sets A we have (using the notation (A.1.1))

$$\mu(A) \le \nu(A^\varepsilon) + \varepsilon.$$

⌟

It can be shown that this is a metric and it generates the weak topology. In computer science, it has been reinvented by the name of "earth mover distance". The following important result helps visualize it:

Proposition A.2.40 (Coupling Theorem, see [49]) *Let μ, ν be two probability measures over a complete separable metric space $\mathbf{X}$ with $\rho(\mu, \nu) \le \varepsilon$. Then there is a probability measure $\mathbf{P}$ on the space $\mathbf{X} \times \mathbf{X}$ with marginals μ and ν such that for a pair of random variables ξ, η having joint distribution $\mathbf{P}$ we have*

$$\mathsf{P}\{d(\xi, \eta) > \varepsilon\} \le \varepsilon.$$

Since weak topology has the second countability property, the metric space defined by the distance $\rho(\cdot, \cdot)$ is separable. This can also be seen directly: let us define a dense set in weak topology.

Definition A.2.41 For each point x, let us define by δ_x the measure which concentrates its total weight 1 in point x. Let D be a countable everywhere dense set of points in X. Let $D_{\mathcal{M}}$ be the set of finite convex rational combinations of measures of the form δ_{x_i} where $x_i \in D$, that is those probability measures that are concentrated on finitely many points of D and assign rational values to them. ⌟

It can be shown that $D_{\mathcal{M}}$ is everywhere dense in the metric space $(\mathcal{M}(X), \rho)$; so, this space is separable. It can also be shown that $(\mathcal{M}(X), \rho)$ is complete. Thus, a measure can be given as the limit of a Cauchy sequence of elements $\mu_1, \mu_2, \ldots$ of $D_{\mathcal{M}}$.

The definition of the Prokhorov distance uses quantification over all Borel sets. However, in an important simple case, it can be handled efficiently.

Proposition A.2.42 *Assume that measure ν is concentrated on a finite set of points $S \subset X$. Then the condition $\rho(\nu, \mu) < \varepsilon$ is equivalent to the finite set of conditions*

$$\mu(A^{\varepsilon}) > \nu(A) - \varepsilon \tag{A.2.7}$$

for all $A \subset S$.

Another useful distance for measures over a bounded space is the Wasserstein distance.

Definition A.2.43 Over a bounded metric space, we define the Wasserstein distance by

$$W(\mu, \nu) = \sup_{f \in \mathrm{Lip}_1(X)} |\mu f - \nu f|.$$

⌟

The Prokhorov and Wasserstein metrics are equivalent.

Proposition A.2.44 *The Prokhorov and Wasserstein metrics are equivalent: the identity function creates a uniformly continuous homeomorphism between the two metric spaces.*

Proof. Let $M = \sup_{x,y \in X} d(x, y)$. The proof actually shows for $\varepsilon < 1$:

$$\rho(\mu, \nu) \leq \varepsilon \Rightarrow W(\mu, \nu) \leq (M + 1)\varepsilon,$$
$$W(\mu, \nu) \leq \varepsilon^2 \Rightarrow \rho(\mu, \nu) \leq \varepsilon.$$

Suppose $\rho(\mu, \nu) \leq \varepsilon < 1$. By the Coupling Theorem (Proposition A.2.40), there are random variables ξ, η over X with a joint distribution, and having respectively the distribution μ and ν, such that $\mathsf{P}\{d(\xi, \eta) > \varepsilon\} \leq \varepsilon$. Then we have

$$\begin{aligned} |\mu f - \nu f| &= |\mathsf{E}\, f(\xi) - \mathsf{E}\, f(\eta)| \leq \mathsf{E}\, |f(\xi) - f(\eta)| \\ &\leq \varepsilon\, \mathsf{P}\{\rho(\xi, \eta) \leq \varepsilon\} + M\, \mathsf{P}\{\rho(\xi, \eta) > \varepsilon\} \leq (M + 1)\varepsilon. \end{aligned}$$

Now suppose $W(\mu, \nu) \leq \varepsilon^2 < 1$. For a Borel set A define $g_{\varepsilon}^{A}(x) = |1 - \rho(x, A)/\varepsilon|^{+}$. Then we have $\mu(A) \leq \mu(g_{\varepsilon}^{A})$ and $\nu g_{\varepsilon}^{A} \leq \nu(A^{\varepsilon})$. Further $\varepsilon g_{\varepsilon}^{A} \in \mathrm{Lip}_1$, and hence $W(\mu, \nu) \leq \varepsilon^2$ implies $\varepsilon\, \mu(g_{\varepsilon}^{A}) \leq \varepsilon\, \nu(\varepsilon g_{\varepsilon}^{A}) + \varepsilon^2$. This concludes the proof by

$$\mu(A) \leq \mu(g_{\varepsilon}^{A}) \leq \nu(\varepsilon g_{\varepsilon}^{A}) + \varepsilon \leq \nu(A^{\varepsilon}) + \varepsilon.$$

□

Relative compactness

Convergence over the set of measures, even over a noncompact space, is sometimes obtained via compactness properties.

Definition A.2.45 A set Π of measures in $(\mathcal{M}(X), \rho)$ is called *sequentially compact* if every sequence of elements of Π contains a convergent subsequence.

A set of Π of measures is called *tight* if for every ε there is a compact set K such that $\mu(K) > 1 - \varepsilon$ for all μ in Π. ⌟

The following important theorem is using our assumptions of separability and completeness of the underlying metric space (X, ρ).

Proposition A.2.46 (Prokhorov) *A set of measures is sequentially compact if and only if it is tight and if and only if its closure is compact in the metric space* $(\mathcal{M}(X), \rho)$.

The following, simplest example is interesting in its own right.

Example A.2.47 The one-element set $\{\mu\}$ is compact and therefore by Prokhorov's theorem tight. Here, tightness says that for each ε a mass of size $1-\varepsilon$ of μ is concentrated on some compact set. ⌟

The following theorem strengthens the observation of this example.

Proposition A.2.48 *A finite measure μ over a separable complete metric space has the property*

$$\mu(B) = \sup\{\mu(K) : \text{ compact } K \subseteq B\}$$

for all Borel sets B.

In case of compact metric spaces, the following known theorem helps.

Proposition A.2.49 *The metric space* $(\mathcal{M}(\mathbf{X}), \rho)$ *of measures is compact if and only if the underlying metric space* (X, d) *is compact.*

So, if (X, d) is not compact then the set of measures is not compact. But still, by Proposition A.2.48, each finite measure is "almost" concentrated on a compact set.

Semimeasures

Let us generalize then notion of semimeasure for the case of general Polish spaces. We use Proposition A.2.36 as a starting point.

Definition A.2.50 A function $f \mapsto \mu f$ defined over the set of all bounded continuous functions is called a *semimeasure* if it has the following properties:

a) Nonnegative: $f \geq 0$ implies $\mu f \geq 0$.

b) Positive homogenous: we have $\mu(af) = a\mu f$ for all nonnegative real $a > 0$.

c) Superadditive: $\mu(f + g) \geq \mu f + \mu g$.

d) Normed: $\mu 1 = 1$.

⌟

Remark A.2.51 The weaker requirement $\mu 1 \leq 1$ suffices, but if we have $\mu 1 \leq 1$ we can always set simply $\mu 1 = 1$ without violating the other requirements. ⌟

Functionals of measures

For a deeper study of randomness tests, we will need to characterize certain functionals of finite measures over a Polish space.

Proposition A.2.52 *Let* $\mathbf{X} = (X, d)$ *be a complete separable metric space with* $\tilde{\mathcal{M}}(\mathbf{X})$ *the set of finite measures over it. A weakly continuous linear function* $F : \tilde{\mathcal{M}}(\mathbf{X}) \to \mathbb{R}$ *can always be written as* $F(\mu) = \mu f$ *for some bounded continuous function* f *over* $\mathbf{X}$.

Proof. Define $f(x) = F(\delta_x)$. The weak continuity of F implies that $f(x)$ is continuous. Let us show that it is bounded. Suppose it is not. Then there is a sequence of distinct points $x_n \in X$, $n = 1, 2, \ldots$ with $f(x_n) > 2^{-n}$. Let μ be the measure $\sum_n 2^{-n}\delta_{x_n}$, then by linearity we have $L(\mu) = \sum_n 2^{-n} f(x_n) > \sum_n 1 = \infty$, which is not allowed.

The function $\mu \mapsto \mu f$ is continuous and coincides with $F(\mu)$ on the dense set of points $D_{\mathcal{M}}$, so they are equal. □

B Constructivity

B.1 Computable topology

B.1.1 Representations

There are several equivalent ways that notions of computability can be extended to spaces like real numbers, metric spaces, measures, and so on. We use the the concepts of numbering (notation) and representation, as defined in [55].

Notation B.1.1 We will denote by $\mathbb{N}$ the set of natural numbers and by $\mathbb{B}$ the set $\{0, 1\}$.

Given a set (an alphabet) Σ we denote by $\Sigma^{\mathbb{N}} = \Sigma^{\mathbb{N}}$ the set of infinite sequences with elements in Σ.

If for some finite or infinite sequences x, y, z, w, we have $z = wxy$ then we write $w \sqsubseteq z$ (w is a *prefix* of z) and $x \triangleleft z$. After [55], let us define the *wrapping function* $\iota : \Sigma^* \to \Sigma^*$ by

$$\iota(a_1 a_2 \cdots a_n) = 110a_1 0a_2 0 \cdots a_n 011. \tag{B.1.1}$$

Note that

$$l(\iota(x)) = (2l(x) + 5) \vee 6. \tag{B.1.2}$$

For strings $x, x_i \in \Sigma^*$, $p, p_i \in \Sigma^{\mathbb{N}}$, $k \geq 1$, appropriate tupling functions $\langle x_1, \ldots, x_k \rangle$, $\langle x, p \rangle$, $\langle p, x \rangle$, and so on can be defined with the help of $\langle \cdot, \cdot \rangle$ and $\iota(\cdot)$. ⌟

Definition B.1.2 Given a countable set C, a *numbering* (or *notation*) of C is a surjective partial mapping $\delta : \mathbb{N} \to C$. Given some finite alphabet $\Sigma \supseteq \{0, 1\}$ and an arbitrary set S, a *representation* of S is a surjective partial mapping $\chi : \Sigma^{\mathbb{N}} \to S$. A *naming system* is a notation or a representation. ⌟

Here are some standard naming systems:

1. id, the identity over Σ^* or $\Sigma^{\mathbb{N}}$.
2. $\nu_{\mathbb{N}}$, $\nu_{\mathbb{Z}}$, $\nu_{\mathbb{Q}}$ for the set of natural numbers, integers and rational numbers.

3. $\mathrm{Cf} : \Sigma^{\mathbb{N}} \to 2^{\mathbb{N}}$, the *characteristic function representation* of sets of natural numbers, is defined by $\mathrm{Cf}(p) = \{i : p(i) = 1\}$.
4. $\mathrm{En} : \Sigma^{\mathbb{N}} \to 2^{\mathbb{N}}$, the *enumeration representation* of sets of natural numbers, is defined by $\mathrm{En}(p) = \{n \in \mathbb{N} : 110^{n+1}11 \triangleleft p\}$.
5. For $\Delta \subseteq \Sigma$, $\mathrm{En}_\Delta : \Sigma^{\mathbb{N}} \to 2^{\Delta^*}$, the *enumeration representation* of subsets of Δ^*, is defined by $\mathrm{En}_\Delta(p) = \{w \in \Sigma^* : \iota(w) \triangleleft p\}$.

Using Turing machines with infinite input tapes, work tapes and output tapes, one can define names for all computable functions between spaces that are Cartesian products of terms of the kind Σ^* and $\Sigma^{\mathbb{N}}$. (One wonders whether $\Sigma^* \cup \Sigma^{\mathbb{N}}$ is not needed, since a Turing machine with an infinite input tape may still produce only a finite output. But in this case one can also encode the result into an infinite sequence.) Then, the notion of computability can be transferred to other spaces as follows.

Definition B.1.3 Let $\delta_i : Y_i \to X_i$, $i = 1, 0$ be naming systems of the spaces X_i. Let $f :\subseteq X_1 \to X_0$, $g :\subseteq Y_1 \to Y_0$. We say that function g *realizes* function f if

$$f(\delta_1(y)) = \delta_0(g(y)) \tag{B.1.3}$$

holds for all y for which the left-hand side is defined.

Realization of multi-argument functions is defined similarly. We say that a function $f : X_1 \times X_2 \to X_0$ is $(\delta_1, \delta_2, \delta_0)$-*computable* if there is a computable function $g :\subseteq Y_1 \times Y_2 \to Y_0$ realizing it. In this case, a name for f is naturally derived from a name of g.[1] ⌟

Definition B.1.4 For representations ξ, η, we write $\xi \leq \eta$ if there is a computable partial function $f : \Sigma^{\mathbb{N}} \to \Sigma^{\mathbb{N}}$ with $\xi(x) = \eta(f(x))$. In words, we say that ξ is *reducible* to η, or that f reduces (translates) ξ to η. There is a similar definition of reduction for notations. We write $\xi \equiv \eta$ if $\xi \leq \eta$ and $\eta \leq \xi$. ⌟

B.1.2 Constructive topological space

Let us start with the definition of topology with the help of a subbasis of (possibly empty) open sets.

Definition B.1.5 A *constructive topological space* $\mathbf{X} = (X, \sigma, \nu)$ is a topological space over a set X with a subbasis σ effectively enumerated (not necessarily without repetitions) as a list $\sigma = \{\nu(1), \nu(2), \ldots\}$, and having the T_0 property (thus, every point is determined uniquely by the subset of elements of σ containing it).

[1] Any function g realizing f via (B.1.3) automatically has a certain *extensionality* property: if $\delta_1(y) = \delta_1(y')$ then $g(y) = g(y')$.

By definition, a constructive topological space satisfies the second countability axiom. We obtain a basis

$$\sigma^{\cap}$$

of the space $\mathbf{X}$ by taking all possible finite intersections of elements of σ. It is easy to produce an effective enumeration for $\sigma^{\cap}$ from ν. We will denote this enumeration by $\nu^{\cap}$. This basis will be called the *canonical basis*.

For every nonempty subset Y of the space X, the subspace of X will naturally get the same structure $\mathbf{Y} = (Y, \sigma, \nu)$ defined by $\{V \cap Y : V \in \sigma\}$.

The *product operation* is defined over constructive topological spaces in the natural way. ⌟

Remark B.1.6 The definition of a subspace shows that a constructive topological space is not a "constructive object" by itself, since the set X itself is not necessarily given effectively. For example, any nonempty subset of the real line is a constructive topological space, as a subspace of the real line. ⌟

Examples B.1.7 The following examples will be used later.

1. A discrete topological space, where the underlying set is finite or countably infinite, with a fixed enumeration.
2. The real line, choosing the basis to be the open intervals with rational endpoints with their natural enumeration. Product spaces can be formed to give the Euclidean plane a constructive topology.
3. The real line $\mathbb{R}$, with the subbasis $\sigma_{\mathbb{R}}^{>}$ defined as the set of all open intervals $(-\infty, b)$ with rational endpoints b. The subbasis $\sigma_{\mathbb{R}}^{<}$, defined similarly, leads to another topology. These two topologies differ from each other and from the usual one on the real line, and they are not Hausdorff spaces.
4. This is the constructive version of Example A.1.4.5. Let X be a set with a constructive discrete topology, and $X^{\mathbb{N}}$ the set of infinite sequences with elements from X, with the product topology: a natural enumerated basis is also easy to define.

⌟

Definition B.1.8 Due to the T_0 property, every point in our space is determined uniquely by the set of open sets containing it. Thus, there is a representation $\gamma_{\mathbf{X}}$ of $\mathbf{X}$ defined as follows. We say that $\gamma_{\mathbf{X}}(p) = x$ if $\mathrm{En}_{\Sigma}(p) = \{w : x \in \nu(w)\}$. If $\gamma_{\mathbf{X}}(p) = x$ then we say that the infinite sequence p is a *complete name* of x: it encodes all names of all subbasis elements containing x. From now on, we will call $\gamma_{\mathbf{X}}$ the *complete standard representation of the space* $\mathbf{X}$. ⌟

Remark B.1.9 Here it becomes important that representations are allowed to be partial functions. ⌟

Definition B.1.10 (Constructive open sets) In a constructive topological space $\mathbf{X} = (X, \sigma, \nu)$, a set $G \subseteq X$ is called *constructive open*, or *lower semicomputable open* in set B if there is a computably enumerable set E with $G = \bigcup_{w \in E} \nu^{\cap}(w) \cap B$. It is constructife open if it is constructive open in X. ⌟

In the special kind of spaces in which randomness has been developed until now, constructive open sets have a nice characterization:

Proposition B.1.11 *Assume that the space $\mathbf{X} = (X, \sigma, \nu)$ has the form $Y_1 \times \cdots \times Y_n$ where each Y_i is either Σ^* or $\Sigma^{\mathbb{N}}$. Then a set G is constructive open iff it is open and the set $\{(w_1, \ldots, w_n) : \bigcap_i \nu(w_i) \subset G\}$ is recursively enumerable.*

Proof. The proof is not difficult, but it relies on the discrete nature of the space Σ^* and on the fact that the space $\Sigma^{\mathbb{N}}$ is compact and its basis consists of sets that are open and closed at the same time. □

It is easy to see that if two sets are constructive open then so is their union. The above remark implies that a space having the form $Y_1 \times \cdots \times Y_n$ where each Y_i is either Σ^* or $\Sigma^{\mathbb{N}}$, also the intersection of two recursively open sets is recursively open. We will see that this statement holds, more generally, in all computable metric spaces.

B.1.3 Computable functions

Definition B.1.12 Let $\mathbf{X}_i = (X_i, \sigma_i, \nu_i)$ be constructive topological spaces, and let $f : X_1 \to X_0$ be a function. As we know, f is continuous iff the inverse image $f^{-1}(G)$ of each open set G is open in its domain. Computability is an effective version of continuity: it requires that the inverse image of basis elements is uniformly constructively open. More precisely, $f : X_1 \to X_0$ is *computable* if the set

$$\bigcup_{V \in \sigma_0^{\cap}} f^{-1}(V) \times \{V\}$$

is a constructive open subset of $X_1 \times \sigma_0^{\cap}$. Here the basis $\sigma_0^{\cap}$ of $\mathbf{X}_0$ is treated as a discrete constructive topological space, with its natural enumeration.

A partial function is computable if its restriction to the subspace that is its domain, is computable. ⌟

The above definition depends on the enumerations ν_1, ν_0. The following theorem shows that this computability coincides with the one obtained by transfer via the representations $\gamma_{\mathbf{X}_i}$.

Proposition B.1.13 (Hertling) *For $i = 0, 1$, let $\mathbf{X}_i = (X_i, \sigma_i, \nu_i)$ be constructive topological spaces. Then a function $f : X_1 \to X_0$ is computable iff it is $(\gamma_{\mathbf{X}_1}, \gamma_{\mathbf{X}_0})$-computable for the representations $\gamma_{\mathbf{X}_i}$ defined above.*

The notion of computable functions helps define morphisms between constructive topological spaces.

Definition B.1.14 Let us call two spaces X_1 and X_0 *effectively homeomorphic* if there are computable maps between them that are inverses of each other. In the special case when $X_0 = X_1$, we say that the enumerations of subbases ν_0, ν_1 are *equivalent* if the identity mapping is a effective homeomorphism. This means that there are recursively enumerable sets F, G such that

$$\nu_1(v) = \bigcup_{(v,w)\in F} \nu_0^{\cap}(w) \text{ for all } v, \quad \nu_0(w) = \bigcup_{(w,v)\in G} \nu_1^{\cap}(v) \text{ for all } w.$$

⌟

B.1.4 Computable elements and sequences

Let us define computable elements.

Definition B.1.15 Let $\mathbf{U} = (\{0\}, \sigma_0, \nu_0)$ be the one-element space turned into a trivial constructive topological space, and let $\mathbf{X} = (X, \sigma, \nu)$ be another constructive topological space. We say that an element $x \in X$ is *computable* if the function $0 \mapsto x$ is computable. It is easy to see that this is equivalent to the requirement that the set $\{u : x \in \nu(u)\}$ is recursively enumerable. Let $\mathbf{X}_j = (X_j, \sigma_j, \nu_j)$, for $i = 0, 1$ be constructive topological spaces. A sequence $f_i, i = 1, 2, \ldots$ of functions with $f_i : X_1 \to X_0$ is a *computable sequence of computable functions* if $(i, x) \mapsto f_i(x)$ is a computable function. ⌟

It is easy to see that this statement is equivalent to the statement that there is a recursive function computing from each i a name for the computable function f_i. To do this formally, one sometimes using the s-m-n theorem of recursion theory, or calls the method "currification".

The notion of computability can be relativized.

Definition B.1.16 Let $X_j, j = 1, 2$ be two constructive topological spaces (or more general, representations) and let $x_j \in X_j$ be given. We say that x_2 is x_1-*computable* if there is a computable partial function $f : X_1 \to X_2$ with $f(x_1) = x_2$. ⌟

Remark B.1.17 The requirement that f computes y from *every* representation of x makes x-computability a very different requirement of mere Turing reducibility of y to x. We will point out an important implication of this difference later, in

the notion of μ-randomness. It is therefore preferable not to use the term "oracle computation" when referring to computations on representations . ⌟

B.1.5 Semicomputability

Lower semicomputability is a constructive version of lower semicontinuity, as given in Definition A.1.14, but the sets that are required to be open there are required to be constructive open here. The analogue of Proposition A.1.17 and Corollary A.1.19 holds also: a lower semicomputable function is the supremum of simple constant functions defined on basis elements, and a lower semicomputable function defined on a subset can always be extended to one over the whole space.

Definition B.1.18 Let $\mathbf{X} = (X, \sigma, \nu)$ be a constructive topological space. A partial function $f : X \to \overline{\mathbb{R}}_+$ with domain D is called *lower semicomputable* if the set $\{(x, r) : f(x) > r\}$ is a constructive open subset of $D \times \overline{\mathbb{R}}_+$.

We define the notion of x-lower semicomputable similarly to the notion of x-computability. ⌟

Let $\mathbf{Y} = (\overline{\mathbb{R}}_+, \sigma^<_{\mathbb{R}}, \nu^<_{\mathbb{R}})$ be the effective topological space introduced in Example B.1.7.2, in which $\nu^>_{\mathbb{R}}$ is an enumeration of all open intervals of the form $(r, \infty]$ with rational r. The following characterization is analogous to Proposition A.1.15.

Proposition B.1.19 *A function $f : X \to \mathbb{R}$ is lower semicomputable if and only if it is $(\nu, \nu^>_{\mathbb{R}})$-computable.*

As a name of a computable function, we can use the name of the enumeration algorithm derived from the definition of computability, or the name derivable using this representation theorem.

The following example is analogous to Example A.1.16.

Example B.1.20 The indicator function $1_G(x)$ of an arbitrary constructive open set G is lower semicomputable. ⌟

The following fact is analogous to Proposition A.1.17:

Proposition B.1.21 *Let $f_1, f_2, \ldots$ be a computable sequence of lower semicomputable functions (via their names) over a constructive topological space $\mathbf{X}$. Then $\sup_i f_i$ is also lower semicomputable.*

The following fact is analogous to Proposition A.1.18:

Proposition B.1.22 *Let $\mathbf{X} = (X, \sigma, \nu)$ be a constructive topological space with enumerated basis $\beta = \sigma^\cap$ and $f : X \to \overline{\mathbb{R}}_+$ a lower semicomputable function. Then there is a lower semicomputable function $g : \beta \to \overline{\mathbb{R}}_+$ (where β is taken with the discrete topology) with $f(x) = \sup_{x \in \beta} g(\beta)$.*

In the important special case of Cantor spaces, the basis is given by the set of finite sequences. In this case we can also require the function $g(w)$ to be monotonic in the words w:

Proposition B.1.23 *Let $X = \Sigma^{\mathbb{N}}$ be a Cantor space as defined in Example B.1.7.4. Then $f : X \to \overline{\mathbb{R}}_+$ is lower semicomputable if and only if there is a lower semicomputable function $g : \Sigma^* \to \overline{\mathbb{R}}_+$ (where Σ^* is taken as a discrete space) monotonic with respect to the relation $u \sqsubseteq v$, with $f(\xi) = \sup_{u \sqsubseteq \xi} g(u)$.*

Remark B.1.24 In the above representation, we could require the function g to be computable. Indeed, we can just replace $g(w)$ with $g'(w)$ where $g'(w) = \max_{v \sqsubseteq w} g(v, l(w))$, and $g(v, n)$ is as much of $g(v)$ as can be computed in n steps. ⌟

Upper semicomputability is defined analogously:

Definition B.1.25 We will call a closed set *upper semicomputable* (co-recursively enumerable) if its complement is a lower semicomputable (r.e.) open set. ⌟

The proof of the following statement is not difficult.

Proposition B.1.26 *Let $\mathbf{X}_i = (X_i, \sigma_i, \nu_i)$ for $i = 1, 2, 0$ be constructive topological spaces, and let $f : X_1 \times X_2 \to X_0$, and assume that $x_1 \in X_1$ is a computable element.*

1. *If f is computable then $x_2 \mapsto f(x_1, x_2)$ is also computable.*
2. *If $\mathbf{X}_0 = \overline{\mathbb{R}}$, and f is lower semicomputable then $x_2 \mapsto f(x_1, x_2)$ is also lower semicomputable.*

B.1.6 Effective compactness

Compactness is an important property of topological spaces. In the constructive case, however, a constructive version of compactness is frequently needed.

Definition B.1.27 A constructive topological space X with canonical basis β has *recognizeable covers* if the set

$$\{S \subseteq \beta : S \text{ is finite and } \bigcup_{U \in S} U = X\}$$

is recursively enumerable.

A compact space with recognizeable covers will be called *effectively compact*. ⌟

Example B.1.28 Let $\alpha \in [0, 1]$ be a real number such that the set of rationals less than α is recursively enumerable but the set of rationals larger than α are not. (It is known that there are such numbers, for example $\sum_{x \in \mathbb{N}} 2^{-H(x)}$.) Let X be the subspace of the real line on the interval $[0, \alpha]$, with the induced topology. A basis of this topology is the set β of all nonempty intervals of the form $I \cap [0, \alpha]$, where I is an open interval with rational endpoints.

This space is compact, but not effectively so. ⌟

The following is immediate.

Proposition B.1.29 *In an effectively compact space, in every recursively enumerable set of basis elements covering the space one can effectively find a finite covering.*

It is known that every closed subset of a compact space is compact. This statement has a constructive counterpart:

Proposition B.1.30 *Every upper semicomputable closed subset E of an effectively compact space X is also effectively compact.*

Proof. $U_1, \ldots, U_k$ be a finite set covering E. Let us enumerate a sequence V_i of canonical basis elements whose union is the complement of E. If $U_1 \cup \cdots \cup U_k$ covers E then along with the sequence V_i it will cover X, and this will be recognized in a finite number of steps. □

On an effectively compact space, computable functions have computable extrema:

Proposition B.1.31 *Let X be an effectively compact space, let $f(x)$ be lower semicomputable function on X. Then the infimum (which is always a minimum) of $f(x)$ is lower semicomputable uniformly from the definition of X and f.*

Proof. For any rational number r we must be able to recognize that the minimum is greater than r. For each r the set $\{x : f(x) > r\}$ is a lower semicomputable open set. It is the union of an enumerated sequence of canonical basis elements. If it covers X this will be recognized in a finite number of steps. □

It is known that a continuous function map compact sets into compact ones. This statement also has a constructive counterpart.

Proposition B.1.32 *Let X be an effectively compact space, and f a computable function from X into another constructive topological space Y. Then $f(X)$ is effectively compact.*

Proof. Let β_X and β_Y be the enumerated bases of the space X and Y respectively. Since f is computable, there is a recursively enumerable set $\mathcal{E} \subseteq \beta_X \times \beta_Y$ such that $f^{-1}(V) = \bigcup_{(U,V)\in E} U$ holds for all $V \in \beta_Y$. Let $\mathcal{E}_V = \{U : (U, V) \in \mathcal{E}\}$, then $f^{-1}(V) = \bigcup \mathcal{E}_V$.

Consider some finite cover $f(X) \subseteq V_1 \cup \cdots \cup V_k$ with some $V_i \in \beta_Y$. We will show that it will be recognized. Let $\mathcal{U} = \bigcup_i \mathcal{E}_{V_i}$, then $X \subseteq \bigcup \mathcal{U}$, and the whole set $\mathcal{U} \subseteq \beta_X$ is enumerable. By compactness, there is a finite number of elements $U_1, \ldots, U_n \in \mathcal{U}$ with $X \subseteq \bigcup_i U_i$. By effective compactness, every one of these finite covers will be recognized. And any one such recognition will serve to recognize that $V_1, \ldots, V_k$ is a cover of $f(X)$. □

B.1.7 Computable metric space

Following [7], we define a computable metric space as follows.

Definition B.1.33 A *constructive metric space* is a tuple $\mathbf{X} = (X, d, D, \alpha)$ where (X, d) is a metric space, with a countable dense subset D and an enumeration α of D. It is assumed that the real function $d(\alpha(v), \alpha(w))$ is computable. ⌟

As x runs through elements of D and r through positive rational numbers, we obtain the enumeration of a countable basis $\{B(x, r) : x \in D, r \in \mathbb{Q}\}$ (of balls or radius r and center x) of $\mathbf{X}$, giving rise to a constructive topological space $\tilde{\mathbf{X}}$.

Definition B.1.34 Let us call a sequence $x_1, x_2, \ldots$ a *Cauchy* sequence if for all $i < j$ we have $d(x_i, x_j) \leq 2^{-i}$. To connect to the type-2 theory of computability developed above, the *Cauchy-representation* $\delta_{\mathbf{X}}$ of the space is defined in a natural way. ⌟

It can be shown that as a representation of $\tilde{\mathbf{X}}$, it is equivalent to $\gamma_{\tilde{\mathbf{X}}}$: $\delta_{\mathbf{X}} \equiv \gamma_{\tilde{\mathbf{X}}}$.

Examples B.1.35

1. Example A.1.37 is a constructive metric space, with either of the two (equivalent) choices for an enumerated dense set.
2. Consider the metric space of Example A.1.28.6: the Cantor space (X, d). Let $s_0 \in \Sigma$ be a distinguished element. For each finite sequence $x \in \Sigma^*$ let us define the infinite sequence $\xi_x = xs_0s_0\ldots$. The elements ξ_x form a (naturally enumerated) dense set in the space X, turning it into a constructive metric space.

⌟

Let us point out a property of metric spaces that we use frequently.

Definition B.1.36 For balls $B_j = B(x_j, r_j)$, $j = 1, 2$ we will say $B_1 < B_2$ if $d(x_1, x_2) < r_2 - r_1$. In words, we will say that B_1 is *manifestly included* in B_2. This relation is useful since it is an easy way to see $B_1 \subset B_2$. ⌟

The property can be generalized to some constructive topological spaces.

Definition B.1.37 A constructive topological space (X, β, ν) with basis β has the *manifest inclusion property* if there is a relation $b < b'$ among its basis elements with the following properties.

a) $b < b'$ implies $b \subseteq b'$.

b) The set of pairs $b < b'$ is recursively enumerable.

c) For every point x, and pair of basis elements b, b' containing x there is a basis element b'' containing x with $b'' < b, b'$.

We express this relation by saying that b is *manifestly included* in b'. In such a space, a sequence $b_1 > b_2 > \ldots$ with $\bigcap_i b_i = \{x\}$ is called a *manifest representation* of x. ⌟

Note that if the space has the manifest inclusion property then for every pair $x \in b$ there is a manifest representation of x beginning with b.

A constructive metric space has the manifest inclusion property as a topological space, and Cauchy representations are manifest.

Similarly to the definition of a computable sequence of computable functions in Subsection B.1.4, we can define the notion of a computable sequence of bounded computable functions, or the computable sequence f_i of computable Lipschitz functions: the bound and the Lipschitz constant of f_i are required to be computable from i. The following statement shows, in an effective form, that a function is lower semicomputable if and only if it is the supremum of a computable sequence of computable functions.

Proposition B.1.38 *Let* $\mathbf{X}$ *be a computable metric space. There is a computable mapping that to each name of a nonnegative lower semicomputable function* f *assigns a name of a computable sequence of computable bounded Lipschitz functions* f_i *whose supremum is* f.

Proof sketch. Show that f is the supremum of a computable sequence of functions $c_i 1_{B(u_i,r_i)}$ where $u_i \in D$ and $c_i, r_i > 0$ are rational. Clearly, each indicator function $1_{B(u_i,r_i)}$ is the supremum of a computable sequence of computable functions $g_{i,j}$. We have $f = \sup_n f_n$ where $f_n = \max_{i \le n} c_i g_{i,n}$. It is easy to see that the bounds on the functions f_n are computable from n and that they all are in Lip_{β_n} for a β_n that is computable from n. □

The following is also worth noting.

Proposition B.1.39 *In a computable metric space, the intersection of two constructive open sets is constructive open.*

Proof. Let $\beta = \{B(x,r) : x \in D, r \in \mathbb{Q}\}$ be a basis of our space. For a pair (x,r) with $x \in D$, $r \in \mathbb{Q}$, let

$$\Gamma(x,r) = \{(y,s) : y \in D,\ s \in \mathbb{Q},\ d(x,y) + s < r\}.$$

If U is a constructive open set, then there is a computably enumerable set $S_U \subset D \times \mathbb{Q}$ with $U = \bigcup_{(x,r)\in S_U} B(x,r)$. Let $S'_U = \bigcup\{\Gamma(x,r) : (x,r) \in S_U\}$, then we have $U = \bigcup_{(x,r)\in S'_U} B(x,r)$. Now, it is easy to see

$$U \cap V = \bigcup_{(x,r)\in S'_U \cap S'_V} B(x,r).$$

□

The following theorem is very useful.

Theorem B.1.1 *A computable metric space X is* effectively compact *if and only if from each (rational) ε one can compute a finite set of ε-balls covering X.*

Proof. Suppose first that the space is effectively compact. For each ε, let $B_1, B_2, \ldots$ be a list of all canonical balls with radius ε. This sequence covers the space, so already some $B_1, \ldots, B_n$ covers the space, and this will be recognized.

Suppose now that for every rational ε one can find a finite set of ε-balls covering the space. Let $\mathcal{S} \subseteq \beta$ be a finite set of basis elements (balls) covering the space. For each element $G = B(u, r) \in \mathcal{S}$, let $G_\varepsilon = B(u, r-\varepsilon)$ be its ε-interior, and $\mathcal{S}_\varepsilon = \{G_\varepsilon : G \in \mathcal{S}\}$. Then $G = \bigcup \varepsilon > 0 G_\varepsilon$, and $X = \bigcup_{\varepsilon>0} \bigcup \mathcal{S}_\varepsilon$. Compactness implies that there is an $\varepsilon > 0$ such that already $\mathcal{S}_{eps}$ covers the space. Let $B_1, \ldots, B_n$ be a finite set of $\varepsilon/2$-balls $B_i = B(c_i, r_i)$ covering the space that can be computed from ε. Each of these balls B_i intersects one of the the sets $G_\varepsilon = B(u, r-\varepsilon)$, $d(u, c_i) \leq r - \varepsilon/2$. But then $B_i \subseteq B(u, r)$ in a recognizeable way. Once all the relations $d(u, c_i) \leq r - \varepsilon/2$ will be recognized we will also recognize that $\mathcal{S}$ covers the space. □

We can strengthen now Example A.1.37:

Example B.1.40 (Prove!) Let X be an effectively compact computable metric space. Then the metric space $C(X)$ with the dense set of functions $\mathcal{E}(D)$ introduced in Definition A.1.32 is a computable metric space. ⌟

The structure of a constructive metric space will be inherited on certain subsets:

Definition B.1.41 Let $\mathbf{X} = (X, \mathbf{d}, D, \alpha)$ be a constructive metric space, and $G \subset X$ a constructive open subset. Then there is an enumeration α_G of the set $D \cap G$ that creates a constructive metric space $\mathbf{G} = (G, \mathbf{d}, D, \alpha_G)$. ⌟

Remark B.1.42 An arbitrary subset of a constructive metric space will inherit the constructive topology. It will also inherit the metric, but not necessarily the structure of a constructive metric space. Indeed, first of all it is not necessarily a complete metric space. It also does not inherit an enumerated dense subset. ⌟

B.2 Constructive measure theory

The basic concepts and results of measure theory are recalled in Section A.2. For the theory of measures over metric spaces, see Subsection A.2.6. We introduce a certain fixed, enumerated sequence of Lipschitz functions that will be used

frequently. Let $\mathcal{E}$ be the set of functions introduced in Definition A.1.32. The following construction will prove useful later.

Proposition B.2.1 *All bounded continuous functions can be obtained as the limit of an increasing sequence of functions from the enumerated countable set $\mathcal{E}$ of bounded computable Lipschitz functions introduced in* (A.1.3)*.*

The proof is routine.

B.2.1 Space of measures

Let $\mathbf{X} = (X, d, D, \alpha)$ be a computable metric space. In Subsection A.2.6, the space $\mathcal{M}(\mathbf{X})$ of measures over $\mathbf{X}$ is defined, along with a natural enumeration $\nu = \nu_{\mathcal{M}}$ for a subbasis $\sigma = \sigma_{\mathcal{M}}$ of the weak topology. This is a constructive topological space $\mathbf{M}$ which can be metrized by introducing, as in A.2.6, the *Prokhorov distance* $\rho(\mu, \nu)$. Recall that we defined $D_{\mathbf{M}}$ as the set of those probability measures that are concentrated on finitely many points of D and assign rational values to them. Let $\alpha_{\mathbf{M}}$ be a natural enumeration of $D_{\mathbf{M}}$. Then

$$(\mathcal{M}, \rho, D_{\mathbf{M}}, \alpha_{\mathbf{M}}) \tag{B.2.1}$$

is a computable metric space whose constructive topology is equivalent to $\mathbf{M}$. Let $U = B(x, r)$ be one of the balls in $\mathbf{X}$, where $x \in D_{\mathbf{X}}$, $r \in \mathbb{Q}$. The function $\mu \mapsto \mu(U)$ is typically not computable, since as mentioned in (A.2.6), it is not even continuous. The situation is better with $\mu \mapsto \mu f$. The following theorem is the computable strengthening of part of Proposition A.2.37:

Proposition B.2.2 *Let $\mathbf{X} = (X, d, D, \alpha)$ be a computable metric space, and let $\mathbf{M} = (\mathcal{M}(\mathbf{X}), \sigma, \nu)$ be the constructive topological space of probability measures over $\mathbf{X}$. If the function $f : \mathbf{X} \to \mathbb{R}$ is bounded and computable then $\mu \mapsto \mu f$ is computable.*

Proof sketch. To prove the theorem for bounded Lipschitz functions, we can invoke the Strassen coupling theorem A.2.40.

The function f can be obtained as a limit of a computable monotone increasing sequence of computable Lipschitz functions $f_n^>$, and also as a limit of a computable monotone decreasing sequence of computable Lipschitz functions $f_n^<$. In step n of our computation of μf, we can approximate $\mu f_n^>$ from above to within $1/n$, and $\mu f_n^<$ from below to within $1/n$. Let these bounds be $a_n^>$ and $a_n^<$. To approximate μf to within ε, find a stage n with $a_n^> - a_n^< + 2/n < \varepsilon$. □

Using Example B.1.40, we can extend this as follows:

Proposition B.2.3 (Prove!) *Let $\mathbf{X}$ be an effectively compact metric space (and thus $C(\mathbf{X})$ is a computable metric space). Then the mapping $(\mu, f) \mapsto \mu f$ over $\mathcal{M}(\mathbf{X}) \times C(\mathbf{X})$ is computable.*

Talking about open sets, only a weaker statement can be made.

Proposition B.2.4 *Let $G \subseteq X$ be a constructive open set. The function $\mu \mapsto \mu(G)$ is lower semicomputable.*

Remark B.2.5 Using the notion of an enumerative lattice defined by Hoyrup and Rojas, one can make this a statement about the two-argument function $(\mu, G) \to \mu(G)$. ⌟

It is known that if our metric space $\mathbf{X}$ is compact then so is the space $\mathcal{M}(\mathbf{X})$ of measures. This can be strengthened:

Proposition B.2.6 (Prove!) *If a complete computable metric space $\mathbf{X}$ is effectively compact then $\mathcal{M}(\mathbf{X})$ is also effectively compact.*

B.2.2 Computable and semicomputable measures

A measure μ is called *computable* if it is a computable element of the space of measures. Let $\{g_i\}$ be the set of bounded Lipschitz functions over $\mathbf{X}$ introduced in Definition A.1.32.

Proposition B.2.7 *Measure μ is computable if and only if so is the function $i \mapsto \mu g_i$.*

Proof. The "only if" part follows from Proposition B.2.2. For the "if" part, note that in order to trap μ within some Prokhorov neighborhood of size ε, it is sufficient to compute μg_i within a small enough δ, for all $i \leq n$ for a large enough n. □

A non-computable density function can lead to a computable measure:

Example B.2.8 Let our probability space be the set $\mathbb{R}$ of real numbers with its standard topology. Let $a < b$ be two computable real numbers. Let μ be the probability distribution with density function $f(x) = \frac{1}{b-a} 1_{[a,b]}(x)$ (the uniform distribution over the interval $[a, b]$). Function $f(x)$ is not computable, since it is not even continuous. However, the measure μ is computable: indeed, $\mu g_i = \frac{1}{b-a} \int_a^b g_i(x)dx$ is a computable sequence, hence Proposition B.2.7 implies that μ is computable. ⌟

The following theorem compensates somewhat for the fact mentioned earlier, that the function $\mu \mapsto \mu(U)$ is generally not computable.

Proposition B.2.9 *Let μ be a finite computable measure. Then there is a computable map h with the property that for every bounded computable function f with $|f| \leq 1$ with the property $\mu(f^{-1}(0)) = 0$, if w is the name of f then $h(w)$ is the name of a program computing the value $\mu\{x : f(x) < 0\}$.*

Proof. Straightforward. □

Can we construct a measure just using the pattern of Proposition B.2.7? Suppose that there is a computable function $(i, j) \mapsto m_i(j)$ with the following properties:

a) $i < j_1 < j_2$ implies $|m_i(j_1) - m_i(j_2)| < 2^{-j_1}$.

b) For all n, there is a probability measure μ_n with $m_i(n) = \mu_n g_i$ for all $i < n$.

Thus, the sequences converge, and for each n, all values $m_i(n)$ for $i \leq n$ are consistent with coming from a probability measure ν_n assigning this value to g_i. Is there a probability measure μ with the property that for each i we have $\lim_j m_i(j) = \mu g_i$? Not necessarily, if the space is not compact.

Example B.2.10 Let $X = \{1, 2, 3, \dots\}$ with the discrete topology. Define a probability measure μ_n with $\mu_n g_i = 0$ for $i < n$ and otherwise arbitrarily. Since we only posed $n - 1$ linear conditions on a finite number of variables, it is easy to see that such a μ_n exists. Then define $m_i(n) = \mu_n(i)$ for all i.

Now all the numbers $m_i(n)$ converge with n to 0, but $\mu = 0$ is not a probability measure. ⌟

To guarantee that the sequences $m_i(j)$ indeed define a probability measure, progress must be made, for example, in terms of the narrowing of Prokhorov neighborhoods.

B.2.3 Random transitions

Consider random transitions now.

Definition B.2.11 (Computable kernel) Let $\mathbf{X}, \mathbf{Y}$ be computable metric spaces, giving rise to measurable spaces with σ-algebras $\mathcal{A}, \mathcal{B}$ respectively. Let $\Lambda = \{\lambda_x : x \in X\}$ be a probability kernel from X to Y (as defined in Subsection A.2.5). Let $\{g_i\}$ be the set of bounded Lipschitz functions over Y introduced in Definition A.1.32. To each g_i, the kernel assigns a (bounded) measurable function

$$f_i(x) = (\Lambda g_i)(x) = \lambda_x^y g_i(y).$$

We will call the kernel Λ *computable* if so is the assignment $(i, x) \mapsto f_i(x)$. ⌟

When Λ is computable, each function $f_i(x)$ is of course continuous. The measure $\Lambda^* \mu$ is determined by the values $\Lambda^* g_i = \mu(\Lambda g_i)$, which are computable from (i, μ) and so the mapping $\mu \mapsto \Lambda^* \mu$ is computable.

The following example is the simplest case, when the transition is actually deterministic.

Example B.2.12 (Computable deterministic kernel) A computable function $h : X \to Y$ defines an operator Λ_h with $\Lambda_h g = g \circ h$ (as in Example A.2.29). This is a deterministic computable transition, in which $f_i(x) = (\Lambda_h g_i)(x) = g_i(h(x))$ is, of course, computable from (i, x). We define $h^* \mu = \Lambda_h^* \mu$. ⌟

Bibliography

[1] Yevgeniy A. Asarin. *Individual Random Signals: a Complexity Approach*. PhD thesis, Moscow State University, Moscow, Russia, 1988. In Russian. 4.1.1

[2] Ya. M. Barzdin'. The complexity of programs to determine whether natural numbers not greater than n belong to a recursively enumerable set. *Soviet Math. Doklady*, 9:1251–54, 1968. 3.2.2

[3] Ya. M. Barzdin' and R. Freivald. On the prediction of general recursive functions. *Soviet Math. Doklady*, 206:1224–28, 1972. 1.1.2

[4] Charles H. Bennett and Martin Gardner. The random number Omega bids fair to hold mysteries of the universe. *Scientific American*, 242(5):20–34, Nov 1979. 1.1.3

[5] Patrick Billingsley. *Convergence of Probability Measures*. Wiley, 1968. First edition. Second edition 1999. A.1.5, A.1.40, A.2.6

[6] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13:850–864, 1984. 1.1.2

[7] Vasco Brattka and Gero Presser. Computability on subsets of metric spaces. *Theoretical Computer Science*, 305:43–76, 2003. B.1.7

[8] Gregory J. Chaitin. Information-theoretical limitations of formal systems. *Journal of the ACM*, 21:403–424, 1974. 1.1.2, 1.1.3

[9] Gregory J. Chaitin. *Scientific American*, 232(5):47–52, May 1975. 1.1.2, 1.1.3

[10] Gregory J. Chaitin. A theory of program size formally identical to information theory. *J. Assoc. Comput. Mach.*, 22:329–340, 1975. 1.1.3, 1.6.5

[11] Gregory J. Chaitin. Algorithmic entropy of sets. *Computers and Mathem. with Applications*, 2:233–245, 1976. 1.1.3

[12] Gregory J. Chaitin. Algorithmic information theory. *IBM J. Res.& Dev.*, 21:350–359, 1977. 1.1.3

[13] Imre Csiszár and János Körner. *Information Theory*. Academic Press, New York, 1981. 1.6.13, 1.6.15, 3.1.1

[14] Robert Daley. *Math. Syst. Th.*, 9(1):83–94, 1975. 1.1.2

[15] Pierre Simon de Laplace. *A Philosophical Essay on Probabilities*. Dover, 1951. 1.1, 1.1.3

[16] Terrence Fine. *Theories of Probability*. Academic Press, New York, 1973. 1.1.3

[17] Peter Gács. On the symmetry of algorithmic information. *Soviet Math. Dokl.*, 15:1477–1780, 1974. Translation of the Russian version. 1.1.3, 1.6.5, 3.1.5, 3.1.1

[18] Peter Gács. Exact expressions for some randomness tests. *Z. Math. Log. Grdl. M.*, 26:385–394, 1980. There is an earlier, shorter conference version. 1.1.3, 2.3.6

[19] Peter Gács. On the relation between descriptional complexity and algorithmic probability. *Theoretical Computer Science*, 22:71–93, 1983. Short version: Proc. 22nd IEEE FOCS (1981) 296-303. 1.1.3

[20] Peter Gács. Randomness and probability–complexity of description. In Samuel Kotz and Norman L. Johnson, editors, *Encyclopedia of Statistical Sciences, Vol. 7*, pages 551–556. Wiley, New York, 1986. 1.1.3

[21] Peter Gács. Uniform test of algorithmic randomness over a general space. *Theoretical Computer Science*, 341(1-3):91–137, 2005. 4.6.4

[22] Peter Gács and János Körner. Common information is far less than mutual information. *Problems of Control and Inf. Th.*, 2:149–162, 1973. 3.1.1

[23] Peter Gács, John Tromp, and Paul M. B. Vitányi. Algorithmic statistics. *IEEE Transactions on Information Theory*, 47:2443–2463, 2001. arXiv:math/0006233 [math.PR]. Short version with similar title in Algorithmic Learning Theory, LNCS 1968/2000. 3.1, 3.1.2

[24] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the Association for Computing Machinery*, 33(4):792–807, 1986. 1.1.2

[25] Peter Hertling and Klaus Weihrauch. Randomness spaces. In *Proc. of ICALP'98*, volume 1443 of *Lecture Notes in Computer Science*, pages 796–807. Springer, 1998. 4.1.1, 4.1.4

[26] A.N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Inform. Transmission*, 1(1):1–7, 1965. 1.1, 1.1.3

[27] Andrei N. Kolmogorov. *Foundations of the Theory of Probability*. Chelsea, New York, 1956. 1.1

[28] Andrei N. Kolmogorov. A logical basis for information theory and probability theory. *IEEE Transactions on Information Theory*, IT-14:662–664, 1968. 1.1, 1.1.3

[29] Leonid A. Levin. On the notion of a random sequence. *Soviet Math. Dokl.*, 14(5):1413–1416, 1973. 1.1.3, 4.1.1, 4.1.1

[30] Leonid A. Levin. Laws of information conservation (nongrowth) and aspects of the foundations of probability theory. *Problems of Inform. Transm.*, 10(3):206–210, 1974. 1.1.2, 1.1.3, 1.6.5, 3.1.2

[31] Leonid A. Levin. Uniform tests of randomness. *Soviet Math. Dokl.*, 17(2):337–340, 1976. 4.1.1, 2, 4.3.5

[32] Leonid A. Levin. Randomness conservation inequalities: Information and independence in mathematical theories. *Information and Control*, 61(1):15–37, 1984. 1.1.2, 1.1.3, 2.2.3, 3.1.11, 3.1.2, 4.1.1, 1, 2, 4.3.5

[33] Leonid A. Levin. One-way functions and pseudorandom generators. *Combinatorica*, 4, 1987. 1.1.2

[34] Leonid A. Levin and V.V. V'yugin. Invariant properties of information bulks. In *Proceedings of the Math. Found. of Comp. Sci. Conf.*, volume 53 of *Lecture Notes on Comp.Sci.*, pages 359–364. Springer, 1977. 1.1.3

[35] Ming Li and Paul M. B. Vitányi. *Introduction to Kolmogorov Complexity and its Applications (Third edition)*. Springer Verlag, New York, 2008. 1.1.3, 2.2.1

[36] Donald W. Loveland. A variant of the Kolmogorov concept of complexity. *Information and Control*, 12:510, 1969. 1.1.3, 3.2

[37] Per Martin-Löf. The definition of random sequences. *Information and Control*, 9:602–619, 1966. 1.1.3, 4.1.1, 4.1.1, 4.2.3

[38] Wofgang J. Paul, Joel Seiferas, and Janos Simon. An information-theoretical approach to time-bounds for on-line computation. *J. Computer and System Sciences*, 23:108–126, 1981. 1.1.2

[39] David Pollard. *A User's Guide to Measure-Theoretic Probability*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, U.K., 2001. A.2, A.2.5, A.2.6

[40] Claus Peter Schnorr. Zufälligkeit und Wahrscheinlichkeit. In A. Dold and B. Eckmann, editors, *Lecture Notes in Mathematics 218*, pages 1–213. Springer Verlag, New York, 1971. 1.1.3

[41] Claus Peter Schnorr. Process complexity and effective random tests. *J. Comput. Syst. Sci*, 7(4):376–388, 1973. Conference version: STOC 1972, pp. 168-176. 1.1.3

[42] Claus Peter Schnorr. A review of the theory of random sequences. In *Proc.5-th Int.Congr. of Logic, Meth. and Phil. of Sci., London, Ontario*, 1975. 1.1.3

[43] Claus Peter Schnorr. General random sequences and learnable sequences. *J. Symb. Logic*, 42:329–340, 1977. 1.1.3

[44] Raymond J. Solomonoff. A formal theory of inductive inference I. *Information and Control*, 7:1–22, 1964. 1.1.3

[45] Raymond J. Solomonoff. Complexity-based induction systems: Comparisons and convergence theorems. *IEEE Transactions on Information Theory*, IT-24(4):422–432, July 1978. 1.1.2

[46] Robert Solovay. Unpublished manuscript. 1976. 1.1.3, 1.7.2, 1.7.4

[47] Robert Solovay. On random r.e. sets. In A. I. Arruda et al., editor, *Non-Classical Logic, Model Th. and Computability*, pages 283–307. North Holland, 1977. 1.1.3

[48] Tom Stoppard. *Rosencrantz and Guildenstern are Dead*. A play. 1.1

[49] Volker Strassen. The existence of probability measures with given marginals. *Annals of Mathematical Statistics*, 36:423–439, 1965. A.2.40

[50] J. Ville. *Etude Critique de la Notion de Collectif*. Gauthier-Villars, Paris, 1939. 1.1.3

[51] Richard von Mises and H. Geiringer. *The Mathematical Theory of Probability and Statistics*. Academic Press, New York, 1964. 1.1, 1.1.3

[52] Vladimir G. Vovk and V. V. Vyugin. On the empirical validity of the Bayesian method. *Journal of the Royal Statistical Society B*, 55(1):253–266, 1993. 4.5.3

[53] V. V. V'yugin. On Turing-invariant sets. *Soviet Math. Doklady*, 229(4):1090–1094, 1976. 1.1.3

[54] V. V. V'yugin. Ergodic theorems for individual random sequences. *Theoretical Computer Science*, 207(2):343–361, 1998. 4.2.2

[55] Klaus Weihrauch. *Computable Analysis*. Springer, 2000. 4.1.1, B.1.1, B.1.1

[56] D. G. Willis. Computational complexity and probability constructions. *J. ACM*, pages 241–259, April 1970. 1.1.3

[57] Andrew C. Yao. Theory and applications of trapdoor functions. In *IEEE Symp. on Foundations of Computer Science*, pages 80–91, 1982. 1.1.2

[58] Ann Yasuhara. *Recursive Function Theory and Logic*. Academic Press, New York, 1971. 1.1.1

[59] Alexander K. Zvonkin and Leonid A. Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Math. Surveys*, 25(6):83–124, 1970. 1.1.2, 1.1.3, 3.2