CHAPTER 14

Algorithmic statistics

14.1. The framework and randomness deficiency

Generally speaking, mathematical statistics deals with the following problem: there are some experimental data, and we look for a reasonable theory that explains these data (is consistent with these data). It turns out that the notion of complexity is helpful in understanding this problem. This is a topic of *algorithmic statistics*.[1]

Consider the following (simplified) example. A "black box", switched on, has produced a sequence of bits, say, of length 10^6. (This sequence could also be considered as a number between 0 and $2^{1,000,000} - 1$.) What information about the internal structure of the black box could we get by analyzing this sequence? Or, at least, what conjectures about this internal structure look compatible with these data?

Classical statistics is not well suited to this situation. If we had information from several independent copies of our device, or if we could switch on the device many times (and have good reason to believe that the results are independent), or if we had some probabilistic distribution that depends on a parameter and needed to choose the most suitable value of this parameter—in all these cases the statistic would know what to do. But if our experiment cannot be repeated (which is not uncommon in practice, by the way) and we have no a priori information about the family of possible distributions, statistics does not tell us what to do. Indeed, we have a set of all $2^{1,000,000}$ possible outcomes, and no structure on this set, so what can we say about one specific outcome?

Common sense nevertheless supports some conclusions even in this case. For example, if our device produced 10^6 zeros, then many people would think that the device is indeed very simple and can produce only zeros. Similarly, if the sequence was $010101\cdots$ (alternating zeros and ones), people would probably believe that the black box is a simple mechanism of a flip-flop type. And if the sequence had no visible regularities, people would probably think that the device is some kind of random bit generator. So the conclusions could be quite different, and it would be interesting to give some more formal support for our common sense reasoning.

In the first example (a zero string) the "explanation" (hypothesis) is a singleton: we think that perhaps the device can produce only this string. In the second example (and in all similar situations when the device produces a binary string x of a very small complexity) the same explanation looks reasonable: we believe that the device is made just for producing this specific string x. So the set of possibilities

[1] An alternative short introduction to this topic can be found in [**201**] (without proofs). A more detailed exposition that contains some material of this chapter but puts it in a different perspective can be found in a recent survey paper [**202**].

is a singleton $\{x\}$. On the other hand, in the third example (a random-looking sequence) the "explanation set" is the set of all strings.

There are some intermediate examples. Imagine that our device produced a sequence of length 10^6 where the first 500,000 bits are zeros and the second half is a random-looking sequence of length 500,000 without any visible regularities. Then we may guess that the device first produces 500,000 zeros and then switches to another mode and produces 500,000 random bits. Here the explanation set has cardinality $2^{500,000}$ and consists of all strings of length 1,000,000 that start with 500,000 zeros.

The general framework that covers all our examples, can be explained as follows: given a string x, we suggest some finite set A that contains x and can be considered as a reasonable explanation for x. What do we mean by "reasonable"? Here are two natural requirements:

- the set A should be simple (its Kolmogorov complexity $C(A)$ should be small);
- the string x should be a "typical" element of A.

More specifically, Kolmogorov complexity $C(A)$ of a finite set A is the complexity of the list of its elements (written in some fixed order, e.g., sorted in alphabetic order, and encoded by a binary string). It does not depend on the specific ordering (lexicographical or any other computable total ordering) and on the encoding (up to a constant).

The notion of a "typical representative of a set" can also be made more precise using Kolmogorov complexity. Recall that if a set A consists of N elements, then the conditional complexity $C(x\,|\,A)$ of every x in A does not exceed $\log N + O(1)$ (each element can be described by its ordinal number in A—assuming that A is known). For most x in A the complexity $C(x\,|\,A)$ is close to $\log N$, since only very few elements have smaller complexity. Informally speaking, an element x is typical in A if $d(x\,|\,A)$ is negligible.

Let us reformulate this in the following way. Consider a finite set A, an element $x \in A$, and the difference

$$d(x\,|\,A) = \log|A| - C(x\,|\,A).$$

As we have seen, this difference is non-negative (up to $O(1)$). We call it the *randomness deficiency* of x as an element of A. Note that we do not use this formula to define $d(x\,|\,A)$ if x is not in A; in this case $d(x\,|\,A)$ is undefined. (It is also natural to let $d(x\,|\,A)$ be $+\infty$ when $x \notin A$, since in this case the explanation A is completely unsuitable for x.)

An element x is *typical* in A if $d(x\,|\,A)$ is negligible.

345 Prove that for a given A the probability of the event "a randomly chosen element $x \in A$ has deficiency greater than k" does not exceed 2^{-k}.

(Here probability means just the fraction of elements with given property in A.) In fact, to make this statement true, we need to replace $\log|A|$ by $\lfloor \log A \rfloor$; since complexity is defined up to a constant anyway, we are not that pedantic.

Let us note also that the function d (with two arguments x and A) is lower semicomputable (enumerable from below): We can effectively provide more and more precise lower bounds for it, but we cannot say when its value was achieved. (Indeed, function C is upper semicomputable.)

346 Assume that a function $\delta(x \mid A)$ is given, where x is a string and A is a set containing that string and δ has the following properties: (a) δ is lower semicomputable; (b) for every finite set A and for every natural number k the fraction of strings in A with $\delta(x \mid A) > k$ is less than 2^{-k}. Then $\delta(x \mid A) \leqslant d(x \mid A) + O(1)$.

This statement is a direct corollary of a similar statement for conditional Kolmogorov complexity (see Theorem 19 on p. 36). Its meaning is the following. There are different opinions about which elements of a given set are typical and which are not. That is, there exist different methods to measure non-typicality. Assume that we normalize each method so that, after normalization, in each set the fraction of k-non-typical element is less than 2^{-k}. Assume also that we can reveal non-typicality of a given string in a given set provided we have enough time for that (that time can be quite long and not bounded by any total computable function). Then there is the best such method in the sense that the deficiency it reveals is not less than the deficiency revealed by any other method (up to an additive constant).

Randomness deficiency in a finite set is similar to randomness deficiency of an infinite sequence with respect to a probability measure (see Section 3.5). More specifically, it is similar to the maximal probability bounded randomness test. One can also define an analogue of an expectationally bounded randomness test.

347 Let the *prefix randomness deficiency* of a string x in a finite set A be defined as $d_P(x \mid A) = \log_2 |A| - K(x \mid A)$. Show that $d_P(x \mid A)$ is a maximal lower semicomputable function δ of x and A such that $(1/|A|) \sum_{x \in A} 2^{\delta(x \mid A)}$ is at most 1 for all finite sets A.

(*Hint*: Recall that prefix complexity coincides with the negative logarithm of the a priori probability.)

Thus a finite set A is considered a good explanation for x if *it is simple and the randomness deficiency* $d(x \mid A)$ *of* x *in* A *is small.* Those strings having such an explanation are called *stochastic.* Are there non-stochastic strings? This question will be answered in the next section.

Notice that we consider only statistical hypotheses that are uniform distributions over finite sets. In a more general framework one can consider also arbitrary probability distributions over strings (say, with finite supports and rational values to avoid technical problems). For such distributions the randomness deficiency of a string x with respect to a distribution P is defined as $-\log_2 P(x) - C(x \mid P)$ (if $P(x) = 0$, then the deficiency is infinite: for such strings x the hypothesis P is completely unsatisfactory).

For uniform distributions (all elements of a finite set A have probability $1/|A|$), the generalized definition of randomness deficiency coincides with the previous one. Notice that the general case is not very different from the case of uniform distributions:

348 Assume that x is a string of length n and P is a probability distribution (not necessarily uniform) of complexity k such that the randomness deficiency of x with respect to P is at most l. Then there is a set A of complexity at most $k + O(\log(l + n))$ containing x such that the randomness deficiency of x in A is at most $l + O(\log(l + n))$.

(*Hint*: Let $A = \{y \mid P(y) \geqslant p\}$ where p is the probability of x with respect to P rounded to the nearest integer power of 2.)

This problem explains why we are considering uniform distributions only. Let us stress that in the definition of Kolmogorov complexity of a finite set of strings we consider the set as a finite object represented by the list of all its elements in the lexicographical order. An alternative approach is to measure the complexity of a set as the minimal length of a program *enumerating* the set. With this approach the definition of stochastic strings becomes trivial: all strings are stochastic. Indeed for every string x of complexity k one can consider the set S_k of all strings of complexity at most k as an explanation for x. It has $O(2^k)$ elements and hence the randomness deficiency of x in S_k is negligible. On the other hand, we can enumerate this set given k and hence S_k can be enumerated by a program of length $\log k + O(1)$. However, intuitively S_k is not a good "explanation" for x.

In the case of general probability distributions (not only uniform), we also consider a distribution as a finite object represented by the list of all pairs $(x, P(x))$ for x in the support of P and arranged lexicographically. This is why we need the support to be finite and the values to be rational. Alternatively, we could consider infinite supports and uniformly computable values—in that case the explanation would be a program computing the function $x \mapsto P(x)$. It is essential that we do not allow lower semicomputable semimeasures represented by programs that *lower semicompute* them. If we did, then any string would obtain a perfect explanation—the maximal lower semicomputable semimeasure.

Historical remark. The first definition of randomness deficiency was given by Kolmogorov, who used the formula $\log|A| - C(x)$. The formula $\log|A| - C(x\,|\,A)$ used throughout the book is due to [**60**] (note that in [**60**] the prefix complexity is used instead of the plain one, the difference is $O(\log(\text{deficiency}))$). Kolmogorov's randomness deficiency $\log|A| - C(x)$ is less than or equal to the randomness deficiency $\log|A| - C(x\,|\,A)$, and they differ by at most $C(A)$. The two deficiencies may differ that much, e.g., for $A = \{x\}$. Perhaps Kolmogorov was interested only in sets A with negligible complexity, in which case these two deficiencies are close. For sets with large complexity the expression $\log|A| - C(x)$ may have large negative value and hardly makes any sense.

14.2. Stochastic objects

A string x is called (α, β)-stochastic if there is a finite set A containing x with $C(A) \leqslant \alpha$ and $d(x\,|\,A) \leqslant \beta$.

A natural question arises. Consider all strings x of length n and consider α and β of order $O(\log n)$ or $o(n)$, making the complexity of explanations for x much smaller than the length of x. For such α, β, are there non-stochastic strings (i.e., "non-explainable" objects)? An affirmative answer to this question is provided by the following theorem.

THEOREM 248. *Assume that* $2\alpha + \beta < n - O(\log n)$. *Then there is a string of length* n *that is not* (α, β)*-stochastic.*

(The accurate statement is that there is a c such that for all large enough n and all α, β with $2\alpha + \beta < n - c\log n$ there is a string of length n that is not (α, β)-stochastic.)

PROOF. Consider the list of all finite sets of complexity at most α. The Kolmogorov complexity of this list is at most $\alpha + O(\log\alpha) = \alpha + O(\log n)$ (see p. 25).

Ignoring additive error terms of order $O(\log n)$ (here and also further) we will assume that the complexity of the list is less than α.

Remove from the list all sets of cardinality more than $2^{\alpha+\beta}$. The Kolmogorov complexity of the resulting list is also less than α. By construction it has at most 2^α sets and each of them has at most $2^{\alpha+\beta}$ elements. Thus the union of all sets in the list has less than $2^{2\alpha+\beta} < 2^n$ strings. Hence there is a string of length n that does not appear in any set from the list. Let t be the lexicographically first such string. Its complexity is at most α, as it can be found given n and the list.

Let us show that this string (denoted by t in the sequel) is not (α, β)-stochastic. Indeed, assume that it is contained in some set A of complexity at most α. The cardinality of A exceeds $2^{\alpha+\beta}$ since all smaller sets were taken into account by construction. Therefore

$$d(t \mid A) = \log \#A - C(t \mid A) > (\alpha + \beta) - C(t) \geqslant (\alpha + \beta) - \alpha \geqslant \beta$$

(one should also add a reserve of size $c \log n$ to compensate for logarithmic terms that we ignore). □

In the other direction we have the following trivial bound:

THEOREM 249. *If $\alpha + \beta > n + O(\log n)$, all the strings of length n are (α, β)-stochastic.*

PROOF. Indeed, we can split all n-bit strings into 2^α sets of size 2^β. □

As we will see later, the reality is closer to this bound than to the bound of the previous theorem. See Problem 365 on p. 449.

It is natural to ask how often non-stochastic objects appear. For example, what is the fraction of non-stochastic objects among all n-bit strings? It is immediately clear that this fraction does not exceed $2^{-\beta}$: Let A be the set of all n-bit strings, and note that strings with deficiency β or more form only a $2^{-\beta}$-fraction of A.

On the other hand, if $2\alpha + \beta \ll n$, we can extend the reasoning used to prove Theorem 248. Namely, for some h we consider all sets of complexity at most α and cardinality at most $2^{\alpha+\beta+h}$. Then we take the first 2^h elements not covered by these sets; it is possible if $2\alpha + \beta + h < n$. The complexity of those elements is bounded by $\alpha + h$, so its deficiency in any set of size greater than $2^{\alpha+\beta+h}$ exceeds β. These arguments (with $O(\log n)$-corrections needed) prove the following statement:

THEOREM 250. *If $2\alpha + \beta < n - O(\log n)$, then the fraction of n-bit strings that are not (α, β)-stochastic is at least $2^{-2\alpha-\beta-O(\log n)}$.*

Instead of a fraction of non-stochastic strings (i.e., the probability of obtaining such a string by tossing a fair coin), one can ask about their total a priori probability (i.e., the probability of obtaining such a string by a universal randomized algorithm). More formally, let $\mathbf{m}(\mathbf{x})$ be the discrete a priori probability of x as defined in Chapter 4: $\mathbf{m}(x) = 2^{-K(x)+O(1)}$. Then we consider the sum of $\mathbf{m}(x)$ over all x of length n that are not (α, β)-stochastic. The following theorem estimates this sum:

THEOREM 251. *If $2\alpha + \beta < n - O(\log n)$ and $\alpha < \beta - O(\log n)$, then this sum equals $2^{-\alpha+O(\log n)}$.*

PROOF. We need to prove both lower and upper bounds for this sum. The lower bound easily follows from the proof of Theorem 248. Indeed, a non-stochastic string

constructed in that proof had complexity α and therefore its a priori probability is $2^{-\alpha}$ (as usual, we ignore $O(\log n)$ corrections needed, now in the exponent).

To get an upper bound, consider the sum of $\mathbf{m}(x)$ over *all* strings of length n. That sum is a real number $\omega \leqslant 1$. Let $\bar{\omega}$ be the number represented by first α bits in the binary representation of ω.

Consider the following measure P on strings of length n associated with $\bar{\omega}$. Start lower semicomputation of $m(x)$ for all strings x of length n and continue until the sum of all obtained lower bounds for $m(x)$ reaches $\bar{\omega}$. Let $P(x)$ be the lower bound for $\mathbf{m}(x)$ we get at that time. If $\bar{\omega}$ and n are given, we can compute $P(x)$ for all x of length n. Therefore the complexity of P is at most α. The sum of differences between $\mathbf{m}(x)$ and $P(x)$ over all strings of length n is bounded by $2^{-\alpha}$.

As we saw in Problem 348, one can use arbitrary finite probabilistic distribution in the definition of stochasticity (with an $O(\log n)$-change in the parameters), not only the uniform ones. It remains to be shown that the total a priori probability of all strings x that have $d(x\,|\,P) > \beta$ is bounded by $2^{-\alpha}$. Indeed, for those strings we have

$$\log P(x) - C(x\,|\,P) > \beta.$$

The complexity of P is bounded by α and therefore $C(x)$ exceeds $C(x\,|\,P)$ at most by α. Thus we have

$$-\log P(x) - C(x) > \beta - \alpha.$$

We ignore $O(\log n)$-terms, so we can replace plain complexity by prefix complexity:

$$-\log P(x) - K(x) > \beta - \alpha.$$

Prefix complexity can be defined in terms of a priori probability, so we get

$$\log(\mathbf{m}(x)/P(x)) > \beta - \alpha$$

for all x that have deficiency exceeding β with respect to P. By assumption, $\alpha < \beta$ with some safety margin (enough to compensate all the simplifications we made), so we may assume that for all those x we have $P(x) < \mathbf{m}(x)/2$, or $(\mathbf{m}(x) - P(x)) > \mathbf{m}(x)/2$. Recall that the sum of $\mathbf{m}(x) - P(x)$ over *all* x of length n does not exceed $2^{-\alpha}$ by construction of $\bar{\omega}$. Hence the sum of $\mathbf{m}(x)$ over all strings of deficiency (with respect to P) exceeding β is at most $2^{-\alpha+1}$, and this is what we wanted to prove. □

The notion of a stochastic object can be considered as a finite analog of the notion of an ML-random sequence with respect to a computable measure. The following problem expresses this similarity in more formal terms.

349 Assume that a sequence ω is ML-random with respect to some computable measure. Prove that for all n the n-bit prefix of the sequence ω is an $(O(\log n), O(\log n))$-stochastic string. (*Hint*: Use Problem 348.) Conclude that there is an infinite sequence that is not ML-random with respect to any computable measure. (*Hint*: Adding a short prefix does not affect non-stochasticity.)

Historical remarks. The first definition of (α, β)-stochasticity was given by Kolmogorov (the authors learned it from his talk given in 1981 [**83**], but most probably it was formulated earlier in 1970s; the definition appeared in print in [**174**]). Kolmogorov and Shen ([**174**]) used the formula $\log |A| - C(x)$ for randomness deficiency.

The existence of non-stochastic objects (Theorem 248) was noted in [**174**]. The first estimates of the a priori measure for the set of non-stochastic objects appeared in [**210**]. The first tight bound $2^{-\alpha}$ for the a priori measure of (α, β)-non-stochastic

objects is due to Muchnik [**139**, Theorem 10.10], who established it for all (α, β) with $3\alpha + \beta \leqslant n$. Both papers [**210**] and [**139**] used the Kolmogorov formula $\log |A| - C(x)$ for randomness deficiency.

Theorem 251 appears to be new. Note that this theorem and Muchnik's result use incomparable assumptions on the parameters α, β. Besides, Theorem 251 estimates the a priori measure of a larger set than Muchnik's result.

14.3. Two-part descriptions

There is another natural way to estimate the quality of statistical hypotheses. Let us start with the following remark. If a string x belongs to some finite set A, we can specify x in two steps:

- first, we specify A;
- then we specify the ordinal number of x in A (in some natural ordering, say, the lexicographic one).

Therefore, we get $C(x) \leqslant C(A) + \log \#A$ for every element x of an arbitrary finite set A (again with logarithmic precision).

There can be many two-part descriptions of the same string x (with different sets A). Which of them are better? Naturally, we would like to make both parts smaller (by finding a simpler and smaller set A): if we can decrease one of the parameters while not increasing the other one, this is an improvement. But which is better: simple A or small complex A? We can compare the lengths of the resulting two-part descriptions and choose a set A which gives the shorter one. This approach is often called the *Minimum Description Length* principle (MDL).

The following simple observation shows that we can move the information from the first part of the description into its second part (leaving the total length almost unchanged). In this way we make the set smaller (the price we pay is that its complexity increases).

THEOREM 252. *Let x be a string, and let A be a finite set that contains x. Let i be a non-negative integer such that $i \leqslant \log \#A$. Then there exists a finite set A' containing x such that $\#A' \leqslant \#A/2^i$ and $C(A') \leqslant C(A) + i + O(\log \min\{i, C(A)\})$.*

PROOF. List all the elements of A in some (say, lexicographic) order. Then split the list into 2^i parts (first $\#A/2^i$ elements, next $\#A/2^i$ elements etc.; we omit evident precautions for the case when $\#A$ is not a multiple of 2^i). Then let A' be the part with x. To specify A', it is enough to specify A and the part number, which requires at most i bits. (The logarithmic term at the end is needed to form a pair of these two descriptions; it is enough to specify the length of the shorter description.) □

We will use the following convenient (though non-standard) terminology: a set A is called a $(k * l)$-*description* (of every its element) if $C(A) \leqslant k$ and $\log \#A \leqslant l$. Theorem 252 can now be formulated as follows: if some x has a $(k * l)$-description, then for every $i \in [0, l]$ it also has $((k + i + O(\log \min(i, k))) * (l - i))$-description.

For a given string x let us consider the set P_x of all pairs $\langle k, l \rangle$ such that x has a $(k * l)$-description, i.e., there exists a set A containing x with $C(A) \leqslant k$ and $\log \#A \leqslant l$. Obviously, this set is closed upwards and contains with each point all points on the right (with the bigger k) and on the top (with bigger l). The last theorem says that we can also move down-right adding $\langle i, -i \rangle$ (with logarithmic precision).

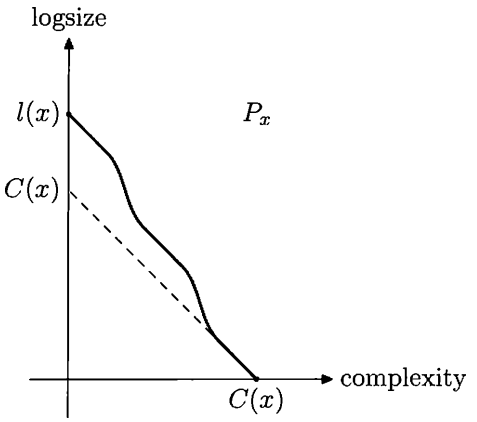


FIGURE 52. The set P_x

We will see that movement in the opposite direction is not always possible. So, having two-part descriptions with the same total length, we should prefer the one with the bigger set (since it always can be converted into others, but not vice versa).

Let us look again at the set P_x for some n-bit string x; see Figure 52. It contains the point $\langle 0, n\rangle$ that corresponds to $A = \mathbb{B}^n$, the set of all n-bit strings (with logarithmic precision). On the other side the set P_x contains the point $\langle C(x), 0\rangle$ that corresponds to the singleton $A = \{x\}$. The boundary of P_x is some curve connecting these two points, and this curve never gets into the triangle $k+s \leqslant C(x)$ and always goes down (when moving from left to right) with slope at least -1 or more, as Theorem 252 says.

This picture raises a natural question: Which boundary curves are possible and which are not? Is it possible, for example, that the boundary goes along the dotted line on Figure 52? The answer is positive: take a random string of the desired complexity and add trailing zeros to achieve the desired length. Then the point $\langle 0, C(x)\rangle$ (the left end of the dotted line) corresponds to the set A of all strings of the same length having the same trailing zeros. We know that the boundary curve cannot go down slower than with slope -1 and that it should end at $\langle C(x), 0\rangle$, therefore it follows the dotted line (with logarithmic precision).

There is a more difficult question: Is it possible that the boundary curve starts from $\langle 0, n\rangle$ and goes with the slope -1 to the very end and then goes down rapidly to $\langle C(x), 0\rangle$? (See Figure 53.) Such a string x, informally speaking, would have essentially only two types of statistical explanations: a set of all strings of length n (and its parts obtained by Theorem 252) and the exact description, the singleton $\{x\}$.

350 Show that such x is not (α, β)-stochastic if α, β are smaller than $C(x)$ and $n - 2C(x)$, respectively.

It turns out that not only are these two opposite cases possible, but also all intermediate curves are possible (assuming they have a bounded slope and are simple enough), if we allow a logarithmic deviation from the prescribed curve.

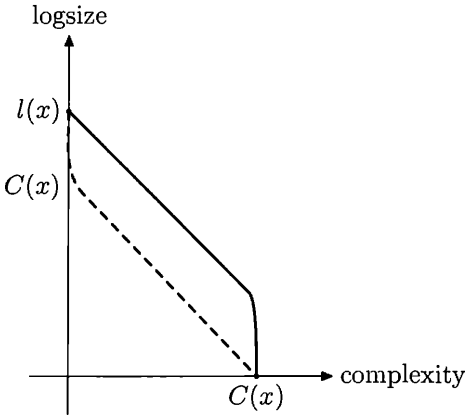


FIGURE 53. Two opposite possibilities for a boundary curve

THEOREM 253. *Let $k \leqslant n$ be two integers, and let $t_0 > t_1 > \cdots > t_k$ be a strictly decreasing sequence of integers such that $t_0 \leqslant n$ and $t_k = 0$; let m be the complexity of this sequence. Then there exists a string x of complexity $k + O(\log n) + O(m)$ and length $n + O(\log n) + O(m)$ for which the boundary curve of P_x coincides with the line $(0, t_0)$-$(1, t_1)$-$\cdots$-(k, t_k) with $O(\log n) + O(m)$-precision: the distance between the set P_x and the set $T = \{\langle i, j\rangle \mid (i < k) \Rightarrow (j > t_i)\}$ is bounded by $O(\log n) + O(m)$.*

(We say that the distance between two sets P and Q is at most ε if P is contained in ε-neighborhood of Q and vice versa.)

PROOF. For every i in the range $0 \cdots k$ we list all the sets of complexity at most i and size at most 2^{t_i}. For a given i the union of all these sets is denoted by S_i. It contains at most 2^{i+t_i} elements. (Here and later we omit constant factors and factors polynomial in n when estimating cardinalities, since they correspond to $O(\log n)$ additive terms for lengths and complexities.) Since the sequence t_i strictly decreases (this corresponds to slope -1 in the picture), the sums $i + t_i$ do not increase, therefore each S_i has at most $2^{t_0} = 2^n$ elements. Therefore, the union of all S_i also has at most 2^n elements (up to a polynomial factor, see above). Therefore, we can find a string of length n (actually $n + O(\log n)$) that does not belong to any S_i. Let x be a first such string in some order (e.g., in lexicographic order).

By construction, the set P_x lies above the curve determined by t_i. So we need to estimate the complexity of x and prove that P_x follows the curve (i.e., that T is contained in the neighborhood of P_x).

Let us start with the upper bound for the complexity of x. The list of all objects of complexity at most k plus the full table of their complexities have complexity $k + O(\log k)$, since it is enough to know k and the number of terminating programs of length at most k. Except for this list, we need to know the sequence $t_0, \ldots, t_k$ whose complexity is m.

For the lower bound, the complexity of x cannot be less than k since all the singletons of this complexity were excluded (via T_k).

It remains to be shown that for every $i \leqslant k$ we can put x into a set A of complexity i (or slightly bigger) and size 2^{t_i} (or slightly bigger). For this we enumerate

a sequence of sets of correct size and show that one of the sets will have the required properties. If this sequence of sets is not very long, the complexity of its elements is bounded. Here are the details.

We start by taking the first 2^{t_i} strings of length n as our first set A. Then we start enumerating all finite sets of complexity at most j and of size at most 2^{t_j} for all $j = 0, \dots, k$, and get an enumeration of all S_j. Recall that x is the first element that does not belong to all such S_j. So, when a new set of complexity at most j and of size at most 2^{t_j} appears, all its elements are included in S_j and removed from A. Until all elements of A are deleted, we have nothing to worry about, since A covers the minimal remaining element. If (and when) all elements of A are deleted, we replace A by a new set that consists of first 2^{t_i} undeleted (yet) strings of length n. Then we wait again until all the elements of this new A are deleted. If (and when) this happens, we take 2^{t_i} first undeleted elements as new A, etc.

The construction guarantees the correct size of the sets and that one of them covers x (the minimal non-deleted element). It remains to estimate the complexity of the sets we construct in this way.

First, to start the process that generates these sets, we need to know the length n (actually something logarithmically close to n) and the sequence $t_0, \dots, t_k$. In total we need $m + O(\log n)$ bits. To specify each version of A, we need to add its version number. So we need to show that the number of different A's that appear in the process is at most 2^i or slightly bigger.

A new set A is created when all the elements of the old A are deleted. Let us distinguish two types of changes of A: the first changes after a new set of complexity j appears with $j \leqslant i$ and the remaining changes. The changes of the first type can happen only $O(2^i)$ times since there are at most $O(2^i)$ sets of complexity at most i. Thus it suffices to bound the number of changes of the second type. For those changes all the elements of A are removed due to elements of S_j with $j > i$. We have at most 2^{j+t_j} elements in S_j. Since $t_j + j \leqslant t_i + i$, the total number of deleted elements only slightly exceeds 2^{t_i+i}, and each set A consists of 2^{t_i} elements, so we get about 2^i changes of A. □

351 Prove that we cannot strengthen Theorem 253 by requiring the distance between the sets P_x and T be $O(\log n)$ (and not $O(\log n) + O(m)$).

(*Hint*: The number of strings of length $n + O(\log n)$ is much smaller than the number of sets T that satisfy the conditions of the theorem.)

352 Prove that there is no algorithm that, given any x, will find the boundary of the set P_x with accuracy $O(\log l(x))$.

Stronger results on non-computability of the boundary of P_x can be found in the paper [**203**].

Theorem 253 shows that the value of the complexity $C(x)$ does not completely describe the properties of x; different strings x of the same complexity can have different boundary curves of P_x. This curve can be considered an infinite-dimensional characterization of x.

To understand this characteristic better, the following notation is useful. The classification of strings according to their complexity can be represented by an increasing sequence of sets $S_0 \subset S_1 \subset S_2 \cdots$, where S_i is the set of all strings having complexity at most i. The sets S_i are enumerable (uniformly in i); the size of S_i is $O(2^i)$.

Now, instead of this linear classification, we have a two-dimensional family $S_{i,j}$ where $S_{i,j}$ is the union of all finite sets A with $C(A) \leqslant i$ and $\log \#A \leqslant j$ (these sets were called the $(i*j)$-descriptions of their elements). We get a two-dimensional table formed by $S_{i,j}$; note that it is monotone along both coordinates, i.e., $S_{i,j}$ increases when i or j increases. Theorem 252 says that this table is (almost) increasing along the diagonal:

$$S_{i,j} \subset S_{i+k,j-k}.$$

(As usual, we ignore logarithmic corrections: one should write

$$S_{i,j} \subset S_{i+k+O(\log k),j-k}$$

instead.)

To understand better the meaning of this two-dimensional stratification, let us look at the equivalent definitions of $S_{i,j}$. As usual, we ignore the logarithmic terms and consider as identical two families S and S' if $S_{i,j} \subset S'_{i+O(\log l),j+O(\log l)}$ where $l = i + j$.

By an *enumerated list* in the following theorem we mean an algorithm that (from time to time) emits binary strings (perhaps, with repetitions); the length of such a list is defined as the number of strings emitted (each string is counted as many times as it was emitted). Condition (c) assumes that the algorithm can produce strings in groups of arbitrary size (different groups produced by the same algorithm may have different sizes).

THEOREM 254. *The following properties of a string x are equivalent in this sense (each of them implies the others with logarithmic change in the parameters):*

(a) *x belongs to $S_{i,j}$ (has an $(i*j)$-description);*

(b) *there exists a simple (=of complexity $O(\log(i+j))$) enumerated list of size at most 2^{i+j} where x appears (for the first time) at least 2^j steps before the end of the list;*

(c) *there exists a simple (=of complexity $O(\log(i+j))$) enumerated list of size at most 2^{i+j} that includes x where strings are produced in at most 2^i groups;*

(d) *in every simple (=of complexity $O(\log(i+j))$) enumerated list that includes all the strings of complexity at most $i+j$, the string x appears (for the first time) at least 2^j steps before the end of the list.*

PROOF. To show that (a) implies (c), assume that (a) is true. Enumerate all sets of complexity at most i and of size at most 2^j. When a new set appears, it forms a new group added to the list. In this way we get at most 2^i groups of size at most 2^j, so the total length of the enumerated list is at most 2^{i+j}. The complexity of the enumeration algorithm is logarithmic since only i and j should be specified.

To get (b) from (a), we should modify the construction slightly and add 2^j arbitrary elements after each portion. The total number of elements increases then by 2^{i+j} and is still acceptable.

On the other hand, (b) easily implies (a): we need to split the list in groups of size 2^j. Then we get at most 2^i groups, and only 2^j last elements are left outside the groups. Therefore, x is covered by some group. Each group is determined by its ordinal number and therefore has complexity i (plus logarithmic term that covers the complexity of the list).

To get (a) from (c), we split each group into pieces of size 2^j (except for one last piece that can be smaller). The number of full pieces is at most 2^i, since the length of the list is at most 2^{i+j}. The same is true for the number of non-full pieces.

So every piece can be specified by its ordinal number, so its complexity does not exceed i.

So the properties (a)–(c) are equivalent (modulo logarithmic change in parameters), and it remains to show that they are equivalent to (d). Evidently, (d) implies (b), so it is enough to show that (a) implies (d).

So let us assume that x is an element of some finite set A that has complexity at most i and size at most 2^j. All elements of A have complexity at most $i + j + O(\log(i + j))$. As usual, we ignore the logarithmic term and hope that the reader can make the necessary corrections.

Assume also that an enumerated list is given that includes all the strings of complexity at most $i + j$. We want to show that x will appear in this list not too close to the end and at least 2^j strings will follow it. Knowing the set A, we may perform the enumeration until all the elements of A appear in the list. Let B be the part of the list enumerated at that moment. The set B is a finite set of complexity at most i (since it is determined by A and the enumerating algorithm, which is assumed to be simple). Now consider the (lexicographically) first 2^j strings outside B. Each of these strings is determined by B (of complexity i) and ordinal number (at most j bits), so they have complexity at most $i+j$. And all these strings should appear in the enumeration after x. □

One could say that we have introduced an additional classification of strings of complexity at most l by measuring the distance to the end of the list. In terms of our two-dimensional stratification, we can speak of an increasing sequence of sets $S_{i,j}$ on the diagonal $i+j = l$. (Strictly speaking, the increasing sequence is obtained only after logarithmic corrections.) Random strings of length $n \leqslant l - O(\log l)$ (i.e., the strings of length n and complexity n) are at the beginning of this classification, having $(l * 0)$-descriptions. At the other end we have (few) strings that have only $(0 * l)$-descriptions.

353 Show that all strings at the end of the enumerated list of strings of complexity at most n (that are followed only by poly(n) strings) are almost equal in the sense that the conditional complexity of one of them given the other one is $O(\log n)$.

One might say that the difference between l and the logarithm of the number of strings after x in the enumerated list of all strings of complexity at most l measures how strange x is. (The equivalence of (b) and (d) guarantees that this measure does not depend significantly on the choice of enumeration.) Random strings of length at most $l - O(\log l)$ are not strange at all, while the strings that are close to the end of the list, have maximal strangeness (close to l). But one should keep in mind the following:

- The strangeness of a given string x of complexity k (that is determined by its position in the enumerated list of all strings of complexity at most k) can decrease significantly if we consider the same x as an element of the list of all strings of complexity at most l for some $l > k$. In fact, each string x determines a function that maps $l \geqslant C(x)$ to the number of strings after x in the enumeration of strings of complexity at most l. It is essentially the same curve we considered before (the boundary curve for P_x) but transformed into other coordinates: for every l we look at the moment when the diagonal line $i + j = l$ gets inside P_x.

- The strangeness of strings x and y can be very different even if $C(x\,|\,y) \approx 0$ and $C(y\,|\,x) \approx 0$ at the same time. (Indeed, if $l > C(x) + O(\log C(x))$, then the shortest description for a string x is random and is not strange even if x were.)

  However, if x and y correspond to each other under a simple computable bijection, this is not possible (see the next problem).

354 Assume that x and y correspond to each other under a bijection computed by a program of complexity t. Prove that if $x \in S_{i,j}$, then $y \in S_{i+O(t),j}$.

Recall that there is a simple computable bijection that maps a string x to a string y if and only if the total complexity of each of those strings conditional to the other one is negligible (see Problem 31 on p. 36).

By very similar arguments as those used to prove Theorem 254, we can show that k_n (and also m_n from Theorem 15 (p. 25)) for different n are closely related:

355 Prove that for all $n' < n$ the string $k_{n'}$ (i.e., the binary expansion of the number $k_{n'}$) is equivalent to the length n' prefix of the string k_n. (Two strings x, y are called equivalent if both conditional complexities $C(x\,|\,y), C(y\,|\,x)$ are $O(\log n)$). Show that strings m_n have a similar property.

(*Hint*: (See[**203**].) For k_n we have to show that given any number T larger than $B(n-s)$ we are able to find all strings of complexity at most n except fewer than 2^s such strings, and the other way around. Given such a T, start an enumeration of strings of complexity at most n and output them in portions of size 2^s. After T steps all the complete portions will appear. Indeed, the number of steps needed to output all complete portions can be computed from the number of complete portions which has at most $n-s$ bits. The number of remaining strings is fewer than 2^s. In the opposite direction, given a list of strings of complexity at most n except fewer than 2^s such strings, we again start an enumeration of strings of complexity at most n and wait until all the given strings appear in that enumeration. Let T denote the number of steps when it happens. Then any number $t > T$ has complexity at least $n-s$. Indeed, if $C(t) < n-s$, then consider 2^s first strings outside the list. Each of them has complexity at most n, a contradiction. For m_n the arguments are entirely similar.)

The next result generalizes the statement of Problem 39 on p. 40: *If a string x has many descriptions of size k, it has shorter descriptions.* Now we speak about $(i*j)$-descriptions of x, i.e., finite sets containing x that have complexity at most i and cardinality at most 2^j.

THEOREM 255. *Assume that a string x has at least 2^k sets as $(i*j)$-descriptions. Then x has some $(i*(j-k))$-description and even some $((i-k)*j)$-description.*

In this statement we omit (as usual) the logarithmic error terms (the parameters should be increased by $O(\log(i+j+k))$). The word "even" reminds us about Theorem 252 that allows us to convert $(i-k)*j$-descriptions to $i*(j-k)$-descriptions.

PROOF. The first (simpler) statement is an easy consequence of the arguments used in the proof of Theorem 254. Let us enumerate all sets A of complexity at most i and size at most 2^j and see which strings belong to 2^k or more sets (are covered with multiplicity at least 2^k). We have at most $2^{i+j}/2^k$ such elements, i.e., 2^{i+j-k}, and these elements can be enumerated in at most 2^i groups (each new set A may create one new group). So it remains to recall statement (c) of Theorem 254.

To get a stronger second statement, we need to decrease the number of groups in this argument to 2^{i-k} (keeping the number of elements approximately at the same level). It can be done as follows. Again we enumerate sets of complexity at most i and size at most 2^j and look at the strings that are covered many times. But now we also consider the strings that are covered with multiplicity 2^{k-1} (half of the full multiplicity considered before); we call them *candidates*. When an element with full multiplicity appears, we output this element *together with all candidates that exist at that moment.*

In this way we may output elements that will never reach the full multiplicity, but this is not a problem since the total number of emitted elements can increase at most twice compared to our count. The advantage is that the number of groups is now much smaller: after all candidates are emitted, we need at least 2^{k-1} new sets to get a new element with full multiplicity (its multiplicity should increase from 2^{k-1} to 2^k). □

This result has the following important corollary:

THEOREM 256. *If a string x has an $(i*j)$-description A such that $C(A\,|\,x) \geqslant k$, then x has also an $(i*(j-k))$-description and even an $((i-k)*j)$-description.*

Again we omit the logarithmic corrections needed for the exact formulation.

PROOF. Knowing x and the values of i and j (the latter information is of logarithmic size), we can enumerate all $(i*j)$-descriptions of x. Therefore, the complexity of each $(i*j)$-description given x does not exceed the logarithm of the number of descriptions, and if there is an $(i*j)$-description A with large $C(A\,|\,x)$, this means that there are many descriptions, and we can apply the previous theorem. □

This statement shows that the descriptions with optimal parameters (on the boundary of P_x for a given x) are simple relative to x. Which, intuitively speaking, is not surprising at all: If a description contains some irrelevant information (not related to x), it hardly could be optimal.

Historical remarks. The idea of considering two-part descriptions with optimal parameters goes back to Kolmogorov. Theorem 252 was mentioned by Kolmogorov in his talk in 1974 [**82**]. It appeared in print in [**60, 178**]. Possible shapes of the set P_x (Theorem 253) were found in [**203**]. The enumerations of all objects of bounded complexity and their relation to two-part descriptions were studied in [**60**, Section III, E]. Theorem 254, although inspired by [**60**] and [**203**], is presumably new. Theorems 255 and 256 appeared in [**203**].

14.4. Hypotheses of restricted type

In this section we consider the restricted case: the sets (considered as descriptions, or statistical hypotheses) are taken from some family $\mathcal{A}$ that is fixed in advance. (Elements of $\mathcal{A}$ are finite sets of binary strings.) Informally speaking, this means that we have some a priori information about the black box that produces a given string: This string is obtained by a random choice in one of the $\mathcal{A}$-sets, but we do not know in which one.

Before we had no restrictions (the family $\mathcal{A}$ was the family of all finite sets). It turns out that the results obtained so far can be extended (with weaker bounds) to other families that satisfy some natural conditions. Let us formulate these conditions.

(1) The family $\mathcal{A}$ is enumerable. This means that there exists an algorithm that prints elements of $\mathcal{A}$ as lists, with some separators (saying where one element of $\mathcal{A}$ ends and another one begins).

(2) For every n the family $\mathcal{A}$ contains the set $\mathbb{B}^n$ of all n-bit strings.

(3) The exists some polynomial p with the following property: for every $A \in \mathcal{A}$, for every natural n, and for every natural $c < \#A$ the set of all n-bit strings in A can be covered by at most $p(n) \cdot \#A/c$ sets of cardinality at most c from $\mathcal{A}$.

For a string x we denote by $P_x^{\mathcal{A}}$ the set of pairs $\langle i, j\rangle$ such that x has $(i * j)$-description *that belongs to* A. The set $P_x^{\mathcal{A}}$ is a subset of P_x defined earlier; the bigger $\mathcal{A}$ is, the bigger is $P_x^{\mathcal{A}}$. The full set P_x is $P_x^{\mathcal{A}}$ for the family $\mathcal{A}$ that contains all finite sets.

Assume that the family $\mathcal{A}$ has properties (1)–(3). Then for every string x the set $P_x^{\mathcal{A}}$ has properties close to the properties of P_x proved earlier. Namely, for every string x of length n the following is true:

- The set $P_x^{\mathcal{A}}$ contains a pair that is $O(\log n)$-close to $\langle 0, n\rangle$. Indeed, property (2) guarantees that the family $\mathcal{A}$ contains the set $\mathbb{B}^n$ that is an $(O(\log n) * n)$-description of x.
- The set $P_x^{\mathcal{A}}$ contains a pair that is $O(1)$-close to $\langle C(x), 0\rangle$. Indeed, condition (3) applied to $c = 1$ and $A = \mathbb{B}^n$ says that every singleton belongs to A, therefore each string has a $((C(x) + O(1)) * 0)$-description.
- The adaptation of Theorem 252 is true: if $\langle i.j\rangle \in P_x^{\mathcal{A}}$, then

$$\langle i + k + O(\log n), j - k\rangle \in P_x^{\mathcal{A}}$$

for every $k \leqslant j$. (Recall that n is the length of x.) Indeed, assume that x has an $(i * j)$-description $A \in \mathcal{A}$. For a given k we enumerate $\mathcal{A}$ until we find a family of $p(n)2^k$ sets of size $2^{-k}\#A$ (or less) in $\mathcal{A}$ that covers all strings of length n in A. Such a family exists due to (3), and p is the polynomial from (3). The complexity of the set that covers x does not exceed $i + k + O(\log n + \log k)$, since this set is determined by A, n, k and the ordinal number of the set in the cover. We may assume without loss of generality that $k \leqslant n$, otherwise $\{x\}$ can be used as an $((i + k + O(\log n)) * (j - k))$-description of x. So the term $O(\log k)$ can be omitted.

EXAMPLE. Consider the family $\mathcal{A}$ formed by all balls in Hamming's sense, i.e., the sets $B_{y,r} = \{x \mid l(x) = l(y), d(x, y) \leqslant r\}$ (here $l(u)$ is the length of binary string u and $d(x, y)$ is the Hamming distance between two strings x and y of the same length). The parameter r is called the *radius* of the ball and y is its *center*. Informally speaking, this means that the experimental data were obtained by changing at most r bits in some string y (and all possible changes are equally probable). This assumption could be reasonable if some string y is sent via an unreliable channel. Both parameters y and r are not known to us in advance.

356 Prove that for $r \leqslant n$ the set $\mathbb{B}^n$ of n-bit strings can be covered by $\mathrm{poly}(n)2^n/V$ Hamming balls of radius r, where N stands for the cardinality of such a ball (i.e., $V = 1 + n + \cdots + \binom{n}{r}$).

(*Hint*: Consider N balls of radius r whose centers are randomly chosen in $\mathbb{B}^n$. For a given x, the probability of not being covered by any of them equals $(1 - V/2^n)^N < e^{-VN/2^n}$. For $N = n \ln 2 \cdot 2^n/V$ this upper bound is 2^{-n}, so for this N the probability of leaving some x uncovered is less than 1.)

357 Prove that this family (of all Hamming balls) satisfies conditions (1)–(3) above.

(*Hint* for (3): Let A be a ball of radius a, and let c be a number less than $\#A$. We need to cover A by balls of cardinality c or less. Without loss of generality we may assume that $a \leqslant n/2$. Indeed, if $a > n/2$, then we can cover A by two balls A_0, A_1 of radius $n/2$ (the set of all n-bit strings can be covered by two balls of radius $n/2$, whose centers are the all-zero sequence and all-one sequence). Assuming that the statement holds for A_0 and A_1, we cover both A_0 and A_1 and then join the obtained families of balls. As the cardinality of both A_0, A_1 is not more than that of A, we are done.

Let b be the maximal integer in the interval $0 \cdots n/2$ such that the cardinality $|B|$ of a ball of radius b does not exceed c. We will cover A by Hamming balls of radius b. When we increase the radius of the ball by one, its size increases at most $n+1$ times. Therefore, $|B| \geqslant c/(n+1)$, and it suffices to cover A by at most $\mathrm{poly}(n)|A|/|B|$ balls of radius b.

Cover all the strings that are at distance at most b from the center of A by one ball of radius b that has the same center as A. Partition the remaining points into spheres of radii $d = b+1, \ldots, a$: the sphere of radius d consists of all strings at Hamming distance exactly d from the center of A. As the number of those spheres is at most n, it suffices, for every $d \in (b, n/2]$, to cover a sphere of radius d by at most $\mathrm{poly}(n)|S|/|B|$ balls of radius b.

Fix d and a sphere S of radius $d \in (b, n/2]$. We will show that for some f a small family of balls whose centers are at distance f from the center of S covers S. Let f be the solution to the equation $b + f(1 - 2b/n) = d$ rounded to the nearest integer. Consider any ball B of radius b whose center is a distance f from the center of S.

We claim that a fraction at least $1/\mathrm{poly}(n)$ of points in B belong to S. Indeed. let x and y denote the centers of S and B, respectively. Let P denote the set of all indexes i from 1 to n where y coincides with x (i.e., $x_i = y_i$), and let Q stand for the complement of P. Choose a set of $(b/n)|P|$ indexes from P and another set of $(b/n)|Q|$ indexes from Q. Then flip the bits of y with chosen indexes. The resulting string y' is at distance $(b/n)|P| + (b/n)|Q| = b$ from y and at distance $f - (b/n)f + (n-f)(b/n) = d$ from x. Thus y' belongs to the intersection of B and S. The number of strings y' that can be obtained in this way equals $\binom{f}{f(b/n)}\binom{n-f}{(n-f)(b/n)}$. Up to a factor $\mathrm{poly}(n)$ this number equals

$$2^{fh(b/n,1-b/n)+(n-f)h(b/n,1-b/n)} = 2^{nh(b/n,1-b/n)}.$$

On the other hand, the cardinality $|B|$ of a ball of radius b is equal to this number as well, up to a factor $\mathrm{poly}(n)$.

Thus every ball B of radius b with center at distance f from x covers at least $|B|/\mathrm{poly}(n)$ of points from S. Choose such a ball B at random. All points $z \in S$ have the same probability of being covered by B. As each ball B covers $|B|/\mathrm{poly}(n)$ of points from S, this probability is at least $|B|/(|S|\,\mathrm{poly}(n))$. Hence there is a polynomial p such that $p(n)|S|/|B|$ random balls of radius b with centers at distance f from x cover S with positive probability.

358 Consider the family $\mathcal{A}$ that consists of all Hamming balls. Prove that there exists a string x for which the set $P_x^{\mathcal{A}}$ is much smaller than the set P_x. (The

exact statement is for some positive ε and for all sufficiently large n there exists a string x of length n such that the distance between $P_x^{\mathcal{A}}$ and P_x exceeds εn.)

(*Hint*: Fix some α in $(0, 1/2)$ and let V be the cardinality of the Hamming ball of radius αn. Find a set E of cardinality $N = 2^n/V$ such that every Hamming ball of radius αn contains at most n points from E. (This property is related to *list decoding* in coding theory. The existence of such a set can be proved by a probabilistic argument: N randomly chosen n-bit strings have this property with positive probability. Indeed, the probability of a random point being in E is an inverse of the number of points, so the distribution is close to Poisson distribution with parameter 1, and tails decrease much faster than 2^{-n} needed.) Since E can be found by an exhaustive search, we can assume that its complexity is $O(\log n)$ and ignore it (and other $O(\log n)$-terms) in the sequel. Now let x be a random element in E, i.e., a string $x \in E$ of complexity about $\log \#E$. The complexity of a ball A of radius αn that contains x is at least $C(x)$, since knowing such a ball and an ordinal number of x in $A \cap E$, we can find x. Therefore x does not have $(\log \#E, \log V)$-descriptions in $\mathcal{A}$. On the other hand, x does have a $(0, \log \#E)$-description if we do not require it to be in $\mathcal{A}$; the set E is such a description. The point $(\log \#E, \log V)$ is above the line $C(A) + \log \#A = \log \#E$, so $P_x^{\mathcal{A}}$ is significantly smaller than P_x.)

359 Describe the set $P_x^{\mathcal{A}}$ for x constructed in the preceding problem.

(*Hint*: The border of the set $P_x^{\mathcal{A}}$ consists of a vertical segment $C(A) = n - \log V$, where $\log \#A \leqslant \log V$, and the segment of slope -1 defined by $C(A) + \log \#A = n$, where $\log V \leqslant \log \#A$.)

Let $\mathcal{A}$ be a family that has properties (1)–(3). We now prove a (weaker) version of Theorem 253 where the precision is only $O(\sqrt{n \log n})$ instead of $O(\log n)$. Note that with this precision the term $O(m)$ in Theorem 253 (which is proportional to the complexity of the boundary curve) is not needed. Indeed, if we draw a curve on a cell paper with cell size $O(\sqrt{n})$ or larger, the curve goes through $O(\sqrt{n})$ cells and can be described by $O(\sqrt{n})$ bits, so we may assume without loss of generality that the complexity of the curve (the sequence t_i in the statement below) is $O(\sqrt{n})$.

THEOREM 257. *Let $k \leqslant n$ be two integers, and let $t_0 > t_1 > \cdots > t_k$ be a strictly decreasing sequence of integers such that $t_0 \leqslant n$ and $t_k = 0$. Then there exists a string x of complexity $k + O(\sqrt{n \log n})$ and length $n + O(\log n)$ for which the distance between the set $P_x^{\mathcal{A}}$ and the set $T = \{(i, j) \mid (i \leqslant k) \Rightarrow (j \geqslant t_i)\}$ is at most $O(\sqrt{n \log n})$.*

PROOF. The proof is similar to the proof of Theorem 253. Let us first recall that proof. We consider the string x that is the lexicographically first string (of suitable length n') that is not covered by any bad set, i.e., by any set of complexity at most i and size at most 2^j, where the pair (i, j) is at the boundary of the set T. The length n' is chosen in such a way that the total number of strings in all bad sets is strictly less than $2^{n'}$. On the other hand, we need good sets that cover x. For every boundary point (i, j) we construct a set $A_{i,j}$ that contains x and has complexity close to i and size 2^j. The set $A_{i,j}$ is constructed in several attempts. Initially $A_{i,j}$ is the set of lexicographically first 2^j strings of length n'. Then we enumerate bad sets and delete all their elements from $A_{i,j}$. At some step, $A_{i,j}$ may become empty. We then fill it with 2^j lexicographically first strings that are not in the bad sets (at the moment). By construction the final $A_{i,j}$ contains the first x that is not in a bad set (since it is the case all the time). And the set $A_{i,j}$ can

be described by the number of changes (plus some small information describing the process as a whole and the value of j). So it is crucial to have an upper bound for the number of changes. How do we get this bound? We note that when $A_{i,j}$ becomes empty, it is filled again, and all the new elements should be covered by bad sets before the new change could happen. Two types of bad sets may appear: small ones (of size less than 2^j) and large ones (of size at least 2^j). The slope of the boundary line for T guarantees that the total number of elements in all small bad sets does not exceed 2^{i+j} (up to a poly(n)-factor), so they may make $A_{i,j}$ empty only 2^i times. And the number of large bad sets is $O(2^i)$, since the complexity of each is bounded by i. (More precisely, we count separately the number of changes for $A_{i,j}$ that are first changes after a large bad set appears, and the number of other changes.)

Can we use the same argument in the new situation? We can generate bad sets as before and have the same bounds for their sizes and the total number of their elements. So the length n' of x can be the same (in fact, almost the same, as we will need now that the union of all bad sets is less than half of all strings of length n'; see below). Note that we now may enumerate only bad sets in $\mathcal{A}$, since $\mathcal{A}$ is enumerable, but we do not even need this condition. What we cannot do is let $A_{i,j}$ be the set of the first non-deleted elements: we need $A_{i,j}$ to be a set from $\mathcal{A}$.

So we now go in the other direction. Instead of choosing x first and then finding a suitable good $A_{i,j}$ that contains x, we construct the sets $A_{i,j} \in \mathcal{A}$ that change in time in such a way that (1) their intersection always contains some non-deleted element (an element that is not yet covered by bad sets) and (2) each $A_{i,j}$ has not too many versions. The non-deleted element in their intersection (in the final state) is then chosen as x.

Unfortunately, we cannot do this for all points (i,j) along the boundary curve. (This explains the loss of precision in the statement of the theorem.) Instead, we construct good sets only for some values of j. These values go down from n to 0 with step $\sqrt{n \log n}$. We select $N = \sqrt{n/\log n}$ points $(i_1, j_1), \dots, (i_N, j_N)$ on the boundary of T; the first coordinates $i_1, \dots, i_N$ form a non-decreasing sequence, and the second coordinates $j_1, \dots, j_N$ split the range $n \cdots 0$ into (almost) equal intervals ($j_1 = n$, $j_N = 0$). Then we construct good sets of sizes at most $2^{j_1}, \dots, 2^{j_N}$, and denote them by $A_1, \dots, A_N$. All these sets belong to the family $\mathcal{A}$. We also let A_0 be the set of all strings of length $n' = n + O(\log n)$; the choice of the constant in $O(\log n)$ will be discussed later.

Let us first describe the construction of $A_1, \dots, A_N$ assuming that the set of deleted elements is fixed. (Then we discuss what to do when more elements are deleted.) We construct A_s inductively (first A_1, then A_2 etc.). As we have said, $\#A_s \leqslant 2^{j_s}$ (in particular, A_N is a singleton), and we keep track of the ratio

$$(\text{the number of non-deleted strings in } A_0 \cap A_1 \cap \dots \cap A_s)/2^{j_s}.$$

For $s = 0$ this ratio is at least $1/2$; this is obtained by a suitable choice of n' (the union of all bad sets should cover at most half of all n'-bit strings). When constructing the next A_s, we ensure that this ratio decreases only by a poly(n)-factor. How? Assume that A_{s-1} is already constructed; its size is at most $2^{j_{s-1}}$. Condition (3) for $\mathcal{A}$ guarantees that A_{s-1} can be covered by $\mathcal{A}$-sets of size at most 2^{j_s}, and we need about $2^{j_{s-1}-j_s}$ covering sets (up to a poly(n)-factor). Now we let A_s be the covering set that contains the maximal number of non-deleted elements in $A_0 \cap \dots \cap A_{s-1}$. The ratio can decrease only by the same poly(n)-factor. In this

way we get

$$\text{(the number of non-deleted strings in } A_0 \cap A_1 \cap \dots \cap A_s) \geqslant \alpha^{-s} 2^{j_s}/2,$$

where α stands for the poly(n)-factor mentioned above.[2]

Up to now we assumed that the set of deleted elements is fixed. What happens when more strings are deleted? The number of the non-deleted elements in $A_0 \cap \dots \cap A_s$ can decrease, and at some point and for some s it can become less than the declared threshold $\nu_s = \alpha^{-s} 2^{j_s}/2$. Then we can find minimal s where this happens and rebuild all the sets $A_s, A_{s+1}, \dots$ (for A_s the threshold is not crossed due to the minimality of s). In this way we update the sets A_s from time to time, replacing them (and all the consequent ones) by new versions when needed.

The problem with this construction is that the number of updates (different versions of each A_s) can be too big. Imagine that after an update some element is deleted, and the threshold is crossed again. Then a new update is necessary, and after this update the next deletion can trigger a new update, etc. To keep the number of updates reasonable, we will ensure that after the update *for all the new sets* A_l (starting from A_s) *the number of non-deleted elements in* $A_0 \cap \dots \cap A_l$ *is twice bigger than the threshold* $\nu_l = \alpha^{-l} 2^{j_l}/2$. This can be achieved if we make the factor α twice as big: since for A_{s-1} we have not crossed the threshold, for A_s we can guarantee the inequality with additional factor 2.

Now let us prove the bound for the number of updates for some A_s. These updates can be of two types: first, when A_s itself starts the update (being the minimal s where the threshold is crossed); second, when the update is induced by one of the previous sets. Let us estimate the number of the updates of the first type. This update happens when the number of non-deleted elements (that was at least $2\nu_s$ immediately after the previous update of any kind) becomes less than ν_s. This means that at least ν_s elements were deleted. How can this happen? One possibility is that a new bad set of complexity at most i_s (a large bad set) appears after the last update. This can happen at most $O(2^{i_s})$-times, since there are at most $O(2^i)$-objects of complexity at most i. The other possibility is the accumulation of elements deleted due to small bad sets, of complexity at least i_s and of size at most 2^{j_s}. The total number of such elements is bounded by $nO(2^{i_s+j_s})$, since the sum $i_l + j_l$ may only decrease as l increases. So the number of updates of A_s not caused by large bad sets is bounded by

$$nO(2^{i_s+j_s})/\nu_s = \frac{O(n2^{i_s+j_s})}{\alpha^{-s}2^{j_s}} = O(n\alpha^s 2^{i_s}) = 2^{i_s+NO(\log n)} = 2^{i_s+O(\sqrt{n\log n})}$$

(recall that $s \leqslant N$, $\alpha = \text{poly}(n)$, and $N \approx \sqrt{n/\log n}$). This bound remains valid if we take into account the induced updates (when the threshold is crossed for the preceding sets: there are at most $N \leqslant n$ these sets, and an additional factor n is absorbed by O-notation).

We conclude that all the versions of A_s have complexity at most

$$i_s + O(\sqrt{n\log n}),$$

since each of them can be described by the version number plus the parameters of the generating process (we need to know n and the boundary curve, whose

[2]Note that for the values of s close to N, the right-hand side can be less than 1; the inequality then claims just the existence of non-deleted elements. The induction step is still possible: the non-deleted element is contained in one of the covering sets.

complexity is $O(\sqrt{n})$ according to our assumption, see the discussion before the statement of the theorem). The same is true for the final version. It remains to take x in the intersection of the final A_s. (Recall that A_N is a singleton, so the final A_N is $\{x\}$.) Indeed, by construction, this x has no bad $(i*j)$-descriptions where (i,j) is on the boundary of T. On the other hand, x has good descriptions that are $O(\sqrt{n\log n})$-close to this boundary and whose vertical coordinates are $\sqrt{n\log n}$-apart. (Recall that the slope of the boundary guarantees that horizontal distance is less than the vertical distance.) Therefore the position of the boundary curve for $P_x^{\mathcal{A}}$ is determined with precision $O(\sqrt{n\log n})$, as required.[3] □

REMARK. In this proof we may use bad sets not only from $\mathcal{A}$. Therefore, the set $P_x^{\mathcal{B}}$ is close to T for every family $\mathcal{B}$ that contains $\mathcal{A}$, and it is not even needed that $\mathcal{B}$ satisfies requirements (1)–(3) itself.

360 Provide the missing details in this argument.

361 (1) Let x be a string of length n and let r be a natural number not exceeding $n/2$. By $C_r(x)$ we denote the minimal (plain) complexity of a string y of the same length n that differs from x in at most r positions. Prove that (with $O(\log n)$ precision) the value of $C_r(x)$ is the minimal i such that x has $(i*\log V(r))$-description that is a Hamming ball. (Here $V(r)$ is the cardinality of a Hamming ball of radius r in $\mathbb{B}^n$.)

(2) Describe all the possible shapes of the function $C_r(x)$ as a function of r (that appear for different x) with precision $O(\sqrt{n\log n})$.

(*Hint*: For every x in $\mathbb{B}^n$ we have $C_0(x) = C(x)$ and $C_n(x) = O(\log n)$. Also we have

$$0 \leqslant C_a(x) - C_b(x) \leqslant \log(V(b)/V(a)) + O(\log n)$$

for every $a < b \leqslant n/2$. On the other hand, for every $k \leqslant n$ and for every function $t\colon \{0,1,\ldots,n/2\}$ such that

$t(0) = k$, $t(n/2) = 0$ and $0 \leqslant t(a) - t(b) \leqslant \log(V(b)/V(a))$ for every $a < b \leqslant n/2$, there exists a string x of length n and complexity $k + O(\sqrt{n\log n})$ such that $C_a(x) = t(a) + O(\sqrt{n\log n})$ for all $a = 0,1,\ldots n/2$.)

We can again look at the error-correcting codes: If a (Kolmogorov-) simple set of codewords has distance d, then for a codeword x in this set the function $C_r(x)$ does not significantly decrease when r increases from 0 to $d/2$ (indeed, the codeword can be reconstructed from the approximate version of it).

Complexity measure $C_r(x)$ was introduced in the paper [**69**]. In [**54**], this notion was generalized to conditional complexity. There are two natural generalizations, uniform and non-uniform ones. The uniform conditional complexity $C^u_{rs}(x\,|\,y)$ is defined as the minimal length of a program that given any string y' at Hamming distance at most s from y outputs a string x' at Hamming distance at most r from x. It is important that x' may depend on y'. The non-uniform conditional complexity $C_{rs}(x\,|\,y)$ is defined as $\max_{y'}\min_{x'} C(x'\,|\,y')$ where x', y' are at Hamming distance at most r, s from x, y, respectively. The difference between the uniform and the non-uniform definitions is the following. In the non-uniform definition the program to transform y' to x' may depend on y' while in the uniform

[3] Now we see why N was chosen to be $\sqrt{n/\log n}$: the bigger N is, the more points on the curve we have, but then the number of versions of the good sets and their complexity increases, so we have some trade-offs. The chosen value of N balances these two sources of errors.

definition the same short program must transform every y' to an x'. This implies that the non-uniform complexity cannot exceed the uniform one. The non-uniform complexity can by much less than the uniform one (see [**54**] for details).

Theorem 254 provided a criterion saying whether a given string has a $(i*j)$-description (unrestricted). It is not clear whether similar criterion could be found for an arbitrary class $\mathcal{A}$ of allowed descriptions. On the other hand, Theorem 255 is (with minimal changes) valid for an arbitrary enumerable family of descriptions; see conditions (1)–(3) on p. 439.

THEOREM 258. *Let $\mathcal{A}$ be an enumerable family of finite sets. Assume that x is a string of length n that has at least 2^k different $(i*j)$-descriptions from $\mathcal{A}$. (Recall that the $(i*j)$-description of x is a finite set of complexity at most i and cardinality at most 2^j containing x.) Then x has some $((i-k)*j)$-description from $\mathcal{A}$.*

Therefore, if $\mathcal{A}$ satisfies also the requirement (3), the string x in this theorem also has an $(i*(j-k))$-description. (See above about the version of Theorem 252 for restricted descriptions.)

As usual, these statements need logarithmic terms to be exact (this means that $O(\log(n+i+j+k))$-terms should be added to the description parameters).

PROOF. Let us enumerate all $(i*j)$-descriptions from $\mathcal{A}$, i.e., finite sets that belong to $\mathcal{A}$, and have cardinality at most 2^j and complexity at most i. For a fixed n, we start a selection process: some of the generated descriptions are marked (=selected) immediately after their generation. This process should satisfy the following requirements: (1) at any moment every n-bit string x that has at least 2^k descriptions (among enumerated ones) belongs to one of the marked descriptions; (2) the total number of marked sets does not exceed $2^{i-k}p(n,k,i,j)$ for some polynomial p. So we need to construct a selection strategy (of logarithmic complexity). We present two proofs: a probabilistic one and an explicit construction.

PROBABILISTIC PROOF. First we consider a finite game that corresponds to our situation. The game is played by two players, whose turn to move alternates. Each player makes 2^i moves. At each move the first player presents some set of n-bit strings, and the second player replies saying whether it *marks* this set or not. The second player loses, if after some moves the number of marked sets exceeds $2^{i-k+1}(n+1)\ln 2$ (this specific value follows from the argument below) or if there exists a string x that belongs to 2^k sets of the first player but does not belong to any marked set.

Since this is a finite game with full information, one of the players has a winning strategy. We claim that the second player can win. If it is not the case, the first player has a winning strategy. We get a contradiction by showing that the second player has a *probabilistic* strategy that wins with positive probability against any strategy of the first player. So we assume that some (deterministic) strategy of the first player is fixed and consider the following simple probabilistic strategy of the second player: every set A presented by the first player is marked with probability $p = 2^{-k}(n+1)\ln 2$.

The expected number of marked sets is $p2^i = 2^{i-k}(n+1)\ln 2$. By Chebyshev's inequality, the number of marked set exceeds the expectation by a factor 2 with probability less than $1/2$. So it is enough to show that the second bad case (after some move there exists x that belongs to 2^k sets of the first player but does not belong to any marked set) happens with probability at most $1/2$.

For that, it is enough to show that for every fixed x the probability of this bad event is at most $2^{-(n+1)}$. The intuitive explanation is simple: if x belongs to 2^k sets, the second player had (at least) 2^k chances to mark a set containing x (when these 2^k sets were presented by the first player), and the probability of missing all these chances is at most $(1-p)^{2^k}$; the choice of p guarantees that this probability is less than $1/2^{-(n+1)}$. Indeed, using the bound $(1-1/x)^x < 1/e$, it is easy to show that

$$(1-p)^{2^k} < e^{-\ln 2(n+1)} = 2^{-(n+1)}.$$

A meticulous reader would say that this argument is not technically correct since the behavior of the first player (and the moment when the next set containing x is produced) depends on the moves of the second player, so we do not have independent events with probability $1-p$ each (as it is assumed in the computation).[4] The formal argument considers for each t the event R_t "after some move of the second player, the string x belongs to at least t sets provided by the first player, but it does not belong to any selected set". Then we prove by induction (over t) that the probability of R_t does not exceed $(1-p)^t$. Indeed, it is easy to see that R_t is a union of several disjoint subsets (depending on the events happening until the first player provides $t+1$st set containing x), and R_{t+1} is obtained by taking a $(1-p)$-fraction in each of them.

CONSTRUCTIVE PROOF. We consider the same game, but now we allow more sets to be selected (replacing the bound $2^{i-k+1}(n+1)\ln 2$ by a bigger bound $2^{i-k}i^2 n \ln 2$), and we also allow the second player to select sets that were produced earlier (not necessarily upon the preceding move of the first player). The explicit winning strategy for the second players performs simultaneously $i-k+\log i$ substrategies (indexed by the numbers $\log(2^k/i), \log(2^k/i)+1, \ldots, i$).

The substrategy number s wakes up once in 2^s moves (when the number of moves already made by the first player is a multiple of 2^s). It forms a family S that consists of 2^s last sets produced by the first player, and the set T that consists of all strings x covered by at least $2^k/i$ sets from S. Then it selects some elements in S in such a way that all $x \in T$ are covered by one of the selected sets. It is done by a greedy algorithm: first take a set from S that covers a maximal part of T, then take the set that covers a maximal number of non-covered elements, etc. How many steps do we need to cover the entire T? Let us show that

$$(i/2^k)n2^s \ln 2$$

steps are enough. Indeed, every element of T is covered by at least $2^k/i$ sets from S. Therefore, some set from S covers at least $\#T2^k/(i2^s)$ elements, i.e., $2^{k-s}/i$-fraction of T. At the next step the non-covered part is multiplied by $(1-2^{k-s}/i)$ again, and after $in2^{s-k}\ln 2$ steps the number of non-covered elements is bounded by

$$\#T(1-2^{k-s}/i)^{in2^{s-k}\ln 2} < 2^n(1/e)^{n\ln 2} = 1,$$

[4]The same problem appears if we observe a sequence of independent trials. Each of them is successful with probability p, and then we select some trials (before they are actually performed, based on the information obtained so far) and ask what is the probability of the event "t first selected trials were all unsuccessful". This probability does not exceed $(1-p)^t$; it can be smaller if the total number of selected trials is fewer than t with positive probability. This scheme was considered by von Mises when he defined random sequences using selection rules.

therefore all elements of T are covered. (Instead of a greedy algorithm one may use a probabilistic argument and show that randomly chosen $in2^{s-k}\ln 2$ sets from S cover T with positive probability; however, our goal is to construct an explicit strategy.)

Anyway, the number of sets selected by a substrategy number s does not exceed

$$in2^{s-k}(\ln 2)2^{i-s} = in2^{i-k}\ln 2,$$

and we get at most $i^2n2^{i-k}\ln 2$ for all substrategies.

It remains to prove that after each move of the second player every string x that belongs to 2^k or more sets of the first player also belongs to some selected set. For the tth move we consider the binary representation of t,

$$t = 2^{s_1} + 2^{s_2} + \cdots, \text{ where } s_1 > s_2 > \cdots.$$

Since x does not belong to the sets selected by substrategies number $s_1, s_2, \ldots$, the multiplicity of x among the first 2^{s_1} sets is less than $2^k/i$, the multiplicity of x among the next 2^{s_2} sets is also less than $2^k/i$, etc. For those j with $2^{s_j} < 2^k/i$, the multiplicity of x among the respective portion of 2^{s_j} sets is obviously less than $2^k/i$. Therefore, we conclude that the total multiplicity of x is less than $i \cdot 2^k/i = 2^k$, and the second player does not need to care about x. This finishes the explicit construction of the winning strategy.

Now we can assume without loss of generality that the winning strategy has complexity at most $O(\log(n+k+i+j))$. (In the probabilistic argument we have proved the existence of a winning strategy, but then we can perform the exhaustive search until we find one; the first strategy found will have small complexity.) Then we use this simple strategy to play against the strategy of the second player which enumerates all $\mathcal{A}$-sets of complexity less than i and size 2^j (or less). The selected sets can be described by their ordinal numbers (among the selected sets), so their complexity is bounded by $i-k$ (with logarithmic precision). Every string that has 2^k different $(i*j)$-descriptions in $\mathcal{A}$ will also have one among the selected sets, and that is what we need. □

As before (for arbitrary sets), this result implies that explanation with minimal parameters are simple with respect to the explaining object:

THEOREM 259. *Let $\mathcal{A}$ be an enumerable family of finite sets. If a string x has an $(i*j)$-description $A \in \mathcal{A}$ such that $C(A|x) < k$, then x has an $((i-k)*j)$-description in $\mathcal{A}$. If the family $\mathcal{A}$ satisfies condition* (3) *on p. 439, then x has also an $(i*(j-k))$-description in $\mathcal{A}$.*

As usual, we omit the logarithmic corrections needed in the exact statement of this result.

Historical remark. All the results from this section, including non-trivial exercises, are from [**204**]. The probabilistic proof of Theorem 258 was independently proposed by Michal Koucký and Andrei Muchnik.

14.5. Optimality and randomness deficiency

We have considered two ways to measure how bad a finite A is as an explanation for a given object x: the first is the *randomness deficiency* that was defined as

$$d(x\,|\,A) = \log \#A - C(x\,|\,A);$$

the second one, which can be called the *optimality deficiency* and is defined as

$$\delta(x \mid A) = \log \#A + C(A) - C(x),$$

shows how far the two-part description of x using A is from the optimum. How are these two numbers related? First let us make an easy observation.

THEOREM 260. *The randomness deficiency of a string x of a finite set A does not exceed its optimality deficiency (with logarithmic precision, as usual; here $l(x)$ stands for the length of x):*

$$d(x \mid A) \leqslant \delta(x \mid A) + O(\log l(x)).$$

PROOF. We need to prove that

$$\log \#A - C(x \mid A) \leqslant \log \#A + C(A) - C(x) + O(\log l(x)).$$

Canceling the term $\log \#A$, we get an inequality

$$C(x) \leqslant C(A) + C(x \mid A) + O(\log l(x)).$$

Its right-hand side is the complexity of the pair $\langle x, A\rangle$ with accuracy $O(\log C(x, A))$, and it is larger than $C(x)$ with accuracy $O(\log C(x \mid A))$. Note that the bound we are proving should hold with $O(\log l(x))$-precision, and $O(\log C(x \mid A)) = O(\log l(x))$. □

This argument shows that the difference between these two deficiencies is close to $C(x, A) - C(x)$, i.e., to $C(A \mid x)$ with precision $O(\log l(x) + \log C(A))$, and this is $O(\log l(x))$ if $C(A) = O(C(x))$. (There is no sense in considering the explanations that are much more complex than the object they try to explain, so we will always assume that $C(A) = O(C(x))$.)

It is easy to give an example of a hypothesis whose optimality deficiency exceeds significantly its randomness deficiency. Let x be a random string of length n, and let B be the set of all strings of length n plus some random string y of length $n - 1$ that is independent of x. Then $C(B \mid x)$ is close to n, and the optimality deficiency is about n, while the randomness deficiency is still small (including y in the set of all strings of length n does not much change the randomness deficiency of x in that set). In this example, the hypothesis B looks bad from the intuitive viewpoint: It contains an irrelevant element y which has nothing in common with the x that we try to explain. Eliminating this y, we improve the hypothesis and make its optimality deficiency close to its randomness deficiency (which is small in both cases).

Recall that we have proved Theorem 256 which shows that the situation in this example is general: If for a given hypothesis B for a string x the difference between the optimality deficiency $\delta(x \mid B)$ and randomness deficiency $d(x \mid B)$ is large (this difference is about $C(B \mid x)$, as we have seen), then one can find another hypothesis A of the same size and of the same (and even smaller by $C(B \mid x)$) complexity such that $\delta(x \mid A)$ does not exceed $d(x \mid B)$.

Therefore, the question whether for a given string x there exists a set A with $C(A) \leqslant \alpha$ and $d(x \mid A) \leqslant \beta$ (asked in the definition of (α, β)-stochasticity), is equivalent (with logarithmic precision) to the question of whether there exists a set A with $C(A) \leqslant \alpha$ and $\delta(x \mid A) \leqslant \beta$. That is, the set P_x contains the same information about x as the set Q_x of pairs $\langle \alpha, \beta\rangle$ for which x is (α, β)-stochastic, but using different coordinates.

362 Let x be an n-bit string of complexity k. Show that the set P_x (see Theorem 253) determines for which α and β the string x is (α, β)-stochastic: this happens iff the pair $(\alpha, C(x) - \alpha + \beta)$ is in P_x or $\alpha > C(x)$ (with logarithmic accuracy).

363 Prove the claim from p. 429: the first inequality of Theorem 249 can be replaced by a weaker inequality $\alpha + \beta < n - O(\log n)$.

(*Hint*: Consider the first string of length n that has no $\alpha * (n - \alpha)$ descriptions (to be precise we need to subtract $O(\log n)$ from the parameters). Its complexity is close to α. The previous problem implies that x is not (α, β)-stochastic.)

364 Prove that if $\alpha + \beta < n - O(\log n)$, then the fraction of non-(α, β)-stochastic strings is at least $2^{-\alpha-\beta-O(\log n)}$.

(*Hint*: Consider the first $2^{n-\alpha-\beta}$ strings of length n (in lexicographic order) that do not have $(\alpha * (n - \alpha))$-descriptions (we omit logarithmic corrections in the parameters). Each of them has complexity at least α and at most $\alpha + n - \alpha - \beta = n - \beta$. The latter implies that for every x in this set the point $(\alpha, C(x) - \alpha + \beta)$ does not belong to P_x.)

365 Prove that the first inequality of Theorem 251 can be replaced by the weaker inequality $\alpha + \beta < n - O(\log n)$.

(*Hint*: The proof of the upper bound remains almost the same: the a priori probability of a string provided by Problem 363 is at least $2^{-\alpha}$. The proof of the lower bound used only the inequality $\alpha < \beta - O(\log n)$.)

366 For every x consider the set Q_x of all pairs (α, β) such that x is (α, β)-stochastic. Characterize possible behaviors of Q_x.

(*Hint*: Let x be an n-bit string of complexity k. Then the set Q_x is upward closed (i.e., $(\alpha, \beta) \in Q_x$ implies $(\alpha', \beta') \in Q_x$ for all $\alpha' \geqslant \alpha$, $\beta' \geqslant \beta$) and contains pairs $(0, n - k)$ and $(k, 0)$ with logarithmic precision (this means that Q_x contains some pairs $(O(\log n), n - k + O(\log n))$ and $(k + O(1), 0)$). On the other hand, let k and n be some numbers, $k \leqslant n$, and let $s_0, \ldots, s_k$ be a sequence of integers such that $n - k \geqslant s_0 \geqslant s_1 \geqslant \cdots \geqslant s_k = 0$. Let m be the complexity of this sequence. Then there exists a string x of length n and complexity $k + O(\log n) + O(m)$ such that Q_x is $O(\log n) + O(m)$ close to the set $S = \{\langle \alpha, \beta \rangle \mid (\alpha \leqslant k) \Rightarrow (\beta \geqslant s_\alpha)\}$.)

367 Assume that for a string x and some α there exists a hypothesis that achieves minimal randomness deficiency among hypotheses of complexity at most α, and its optimality deficiency exceeds its randomness deficiency by γ. Then the boundary of P_x contains a segment of slope -1 that covers the interval $(\alpha - \gamma, \alpha)$ on the horizontal axis.

(*Hint*: Use the stronger statement of Theorem 256.)

368 Let $\mathcal{A}$ be a family of finite sets that satisfies conditions (1)–(3) on p. 439. Prove that for any x and any $\alpha \leqslant C(x)$ the following are equivalent with logarithmic precision:

- there exists a set $A \in \mathcal{A}$ of complexity at most α with $d(x \mid A) \leqslant \beta$;
- there exists a set $A \in \mathcal{A}$ of complexity at most α with $\delta(x \mid A) \leqslant \beta$;
- the point $(\alpha, C(x) - \alpha + \beta)$ belongs to $P_x^{\mathcal{A}}$.

369 Let $\mathcal{A}$ be an arbitrary family of finite sets enumerated by program p. Prove that for every x of length at most n the following statements are equivalent

up to an $O(C(p) + \log C(A) + \log n + \log\log \#A)$-change in the parameters:

- there exists a set $A \in \mathcal{A}$ such that $d(x \mid A) \leqslant \beta$;
- there exists a set $A \in \mathcal{A}$ such that $\delta(x \mid A) \leqslant \beta$.

Historical remarks. The existence of strings of length n and complexity about k that are not $(k, n-k+O(\log n))$-stochastic was first proved in [**60**, Theorem IV.2]. The study of possible shapes of the set Q_x was initiated by V. V'yugin [**211, 212**] using direct arguments (and not the relation between Q_x and P_x). The descriptions of possible shapes of Q_x with accuracy $O(\log n)$ (Problem 366) is due to [**203**], where reduction to the set P_x is used. Problems 367, 368, and 369 go back to [**203, 204**].

14.6. Minimal hypotheses

Fix a string x. We have associated with x the set P_x consisting of all pairs (α, β) such that x has an $(\alpha * \beta)$-description. Those descriptions were considered as "statistical hypotheses to explain x". What do they look like? It turns out that we can identify a more or less explicit class of models such that every model reduces in a sense to a model from that class. This class arises from the proof of Theorem 254.

Let l be some number greater than $C(x)$. Then the list of all strings of complexity at most l contains x. Fix some enumeration of this list (an algorithm that generates all these strings; each appears only once). We assume that this algorithm is simple: its complexity is $O(\log l)$. Let N_l be the number of elements in the list. Consider the binary representation of N_l, i.e., the sum

$$N_l = 2^{s_1} + 2^{s_2} + \cdots + 2^{s_t}, \text{ where } s_1 > s_2 > \cdots > s_t.$$

According to this decomposition, we may split the list itself into groups: first 2^{s_1} elements, next 2^{s_2} elements, etc. The string x belongs to one of these groups. This group (the corresponding finite set) can be considered as a hypothesis for x. In this way we get a family of models for x: each $l > C(x)$ produces some hypothesis, denoted $B_{x,l}$ in the sequel.

The following two theorems prove the promised properties of these models. First, they are minimal, i.e., they lie on the border of the set P_x. Second, each model for x reduces in a sense to one of them.

THEOREM 261. *Assume that x belongs to the part $B_{x,l}$ of size 2^s in this construction. Then this part is an $((l-s)*s)$-description of x and the point $(l-s, s)$ is on the boundary of P_x. (As usual, the exact statement needs a logarithmic correction: this part is an $((l-s+O(\log l))*s)$-description of x and the corresponding point is in the $O(\log l)$-neighborhood of the boundary of P_x.)*

PROOF. To specify this part, it is enough to know its size and the number of elements enumerated before it, i.e., it is enough to know s, l and all bits of N_l except s last bits (i.e., $l-s$ bits). Also we need to know the enumerating algorithm itself, but it has logarithmic complexity (as we assumed). Therefore the complexity of the part is $l-s+O(\log l)$, and the number of elements is 2^s, as we have claimed.

If the point $(l-s, s)$ were far from the boundary and were in P_x together with more than logarithmic neighborhood, then the string x would have much better two-part descriptions (with the same or even smaller total length and with larger size), so Theorem 254(d) would imply that the string x appears in the list earlier (more than 2^s elements follow x in the enumeration), which is impossible in our construction. □

The next result explains in which sense these descriptions are universal. Let x be an arbitrary string, and let A be some finite set that contains x. Let l be the maximal complexity of the elements of A. As before, let us split the strings of complexity at most l (there are N_l of them) into parts corresponding to ones in the binary representation of N_l. Let B be the part that contains x, and let 2^s be its size.

THEOREM 262. *The hypothesis $B = B_{x,l}$ (considered as an explanation for x) is not worse than A in terms of complexity and optimality deficiency:*

(a) $C(B) \leqslant C(A) + O(\log l)$;

(b) $\delta(x \mid B) \leqslant \delta(x \mid A) + O(\log l)$;

(c) $C(B \mid A) \leqslant O(\log l)$ (*the hypothesis B is simple given A*).

PROOF. Knowing A and l, we can enumerate all strings of complexity at most l until we see all the elements of A. At that moment the string x already appears, and it belongs to the part of size 2^s, so there are only $O(2^s)$ strings yet to be discovered (from this part and the smaller parts). Therefore, we know N_l with precision $O(2^s)$, and therefore we know its first $l - s$ bits (with $O(1)$-advice). And this information, together with l and s, determines B. Therefore, $C(B \mid A) \leqslant O(\log l)$, so we have proved (c) and therefore (a).

The statement (b) follows directly from the construction. Indeed, if $C(A) = \alpha$ and $\log \#A = \beta$, then all the strings in A have an $(\alpha * \beta)$-description and complexity at most $\alpha + \beta + O(\log \alpha)$, so their maximal complexity l does not exceed $\alpha + \beta + O(\log \alpha)$. The two-part description we have constructed is an $((l-s) * s)$-description (as the previous theorem shows), so its total length and optimality deficiency do not exceed those of A. □

The relation between parameters of descriptions A and B is illustrated by Figure 54: the dot corresponds to the parameters of A, and the gray area shows the possible parameters of B.

What happens if the initial hypothesis A is already on the boundary of P_x? Does it mean that B has the same parameters as A? Generally, no: the model B may lie on the dashed part of the boundary of the grey area shown in Figure 54. (It is not possible that B is inside the grey area, since in this case A will correspond to the internal point of P_x.)

In other words, assume that the boundary of P_x consists of vertical lines and non-vertical lines with slope -1. Then the left-upper endpoints of non-vertical

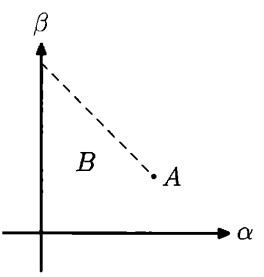


FIGURE 54. The parameters of the hypothesis A and its simplification B

segments correspond to the hypotheses of described type (since for such A the grey area where B resides has only one common point with P_x).

Notice that the information that is contained in these hypotheses, does not really depend on x: the hypothesis B contains the same information as the $(l-s)$-bit prefix of the string N_l. As we have seen in Problem 355 (p. 437), this prefix can be replaced by N_{l-s}, which has the same information as the first $l-s$ bits of Chaitin's Ω number. Thus the larger the complexity of our model is, the more information about Ω it has. This is discouraging, since the number Ω does not depend on x.

It might be that other parameters (than complexity and cardinality) help to distinguish models of the same size and complexity, as explanations for x. The paper [**199**] suggests one such parameter, namely the total complexity A conditional to x. In all our examples intuitively right models for x have small total complexity conditional to x. On the other hand, one can show that models from the universal family from Theorem 261 have large total complexity conditional to some of their members. We omit the proof of this claim, which may be found in [**199**].

Note also that this observation (saying that different hypotheses contain almost the same information) is applicable only to hypotheses of our special type and not to arbitrary hypotheses on the boundary of P_x, as the following example shows. Let x be a random n-bit string. Consider two hypotheses: the set of n-bit strings y that have the same first half as x and the set of n-bit strings y that have the same second half as x. Both hypotheses have small optimality deficiency, but the information contained in them is completely different. (This does not contradict our results above, since the set of all n-bit strings as B has better parameters than both.)

Historical remarks. Cutting the list of all strings of complexity at most k into portions according to the binary expansion of N_k was introduced in [**60**], where it was noticed that for $k = C(x)$ we obtain in this way a model for x with small optimality deficiency. Later in [**203**] models of this type were considered also for $k > C(x)$, and Theorems 261 and 262 were proven.

14.7. A bit of philosophy

There are several philosophical questions related to the task of finding a good two-part description for a given string x. For instance, we can let x be the sequence of all observations about the world made by mankind (encoded in binary) and then consider scientific theories as models A for x. Among those theories we want do identify the right ones. Our criteria are the simplicity of the theory in question (measured by the Kolmogorov complexity of A—the less the complexity is the better), and the "concreteness" or the "explanatory capability" (measured by the size of A—the less the size is, the more concrete the model is, hence the better). One can also recall the ancient philosopher Occam and his razor ("entities must not be multiplied beyond necessity"), which advises choosing the simplest explanation. Or we can look for a scientific theory A such that the randomness deficiency of the data x with respect to A is small ("a good theory should explain all the regularities in the data").

There are also more practical issues related to algorithmic statistics. Kolmogorov complexity can be considered as a theory of "ultimate compression": the

complexity of a string x is the lower bound for its compressed size for compressors without loss of information. The closer to this bound the compressed size is the better the compression method is (for files from a practically important family of files).

This applies to lossy data compression. What about loss compression? Nowadays many compression techniques are used that discard certain not important parts of the information that is being encoded. Such methods allow us to decrease the compressed size below Kolmogorov complexity.

For instance, assume that we are given an old phonograph record that has scratches in random places on the record. These scratches produce peaks on the waveform of the sound (the two-dimensional plot of sound pressure as a function of time). Thus the original information has been distorted. Due to this distortion the Kolmogorov complexity of the record has been much increased (if there are many scratches). However, if we care only about the general impression of playing the record, the exact spots of the scratches are not important. It is enough to store in the compressed file only the general character of the scratches.

In other words, our phonograph record is an element of a large family that consists of all the records with about the same number of scratches of the same type. In this way we obtain a two-part description of the record: the first part is the description of this set (the clean record and statistical parameters of the noise) and the second part identifies the exact spots of the scratches. If our method of compression discards the second part, then after decompression we will get another record. That record will be obtained from the original clean record by adding another noise with the same statistical parameters. One can hope that the audience will not notice the change. Besides, if the decompressing program does not add any noise at all to the clean record, thus "de-noising" the record, then we obtain an even better result (unless of course we are interested in listening to an ancient phonograph instead of listening to music).

The statement of Problem 369 can be interpreted as follows using this analogy. Assume that a string x was obtained from an unknown string y of the same length by adding a noise. That is, for some known natural number r the string x was obtained by a random sampling in a radius-r Hamming ball with the center y. We want to de-noise x and to this end we are looking for a Hamming ball of radius r that provides the minimal length two-part description for x (that is, the Hamming ball of minimal complexity). Assume that we have succeeded and such a ball is found. With high probability the randomness deficiency of x in the original ball is small. By Problem 369 (for the family of all Hamming balls of radius r) the randomness deficiency of x in the ball we have found is small as well. Thus the second part in the found two-part description for x has no useful information. In other words, the center of the ball we have found is a de-noised version of x (in particular, we have also removed the noise present in y).

Here is another example of lossy compression via Kolmogorov complexity. Kolmogorov complexity of a high-resolution picture of a sand-dune is very large, as it identifies the locations of all individual grains of sand, which are random. For a person who looks at that picture, the picture is just a typical element of the set of all similar pictures, where the sand-dune is at the same place, has the same form, and consists of the sand of the same type, while individual sand grains may occupy arbitrary spots. If our compressor stores only the description of this large set

and the decompressing program finds any typical element of that set, the person contemplating the picture will hardly notice any difference.

We should remember that this is just an analogy and we should not expect that mathematical theorems on Kolmogorov complexity of two-part descriptions will be directly applied in practice. One of the reasons for that is our ignoring the computational complexity of decompressing programs and ignoring compressing programs at all. It might be that it is this ignoring that implies paradoxical independence of some minimal models on the string x mentioned earlier.

APPENDIX 1

Complexity and foundations of probability

In this section there are no theorems and no proofs. Instead, we discuss the foundations of probability theory (the connection between probability theory as a part of mathematics, and its applications to the real world), especially the role of the algorithmic information theory, following [**180**].

Probability theory paradox

One often describes the natural sciences framework as follows: A hypothesis is used to predict something, and the prediction is then checked against the observed actual behavior of the system. If there is a contradiction, the hypothesis needs to be changed.

Can we include probability theory in this framework? A statistical hypothesis (say, the assumption of a fair coin) should be then checked against the experimental data (results of coin tossing) and rejected if some discrepancy is found. However, there is an obvious problem: The fair coin assumption says that in a series of, say, 1000 coin tossings all of the 2^{1000} possible outcomes (all 2^{1000} bit strings of length 1000) have the same probability 2^{-1000}. How can we say that some of them contradict the assumption while other do not?

The same paradox can be explained in a different way. Consider a casino that wants to outsource the task of card shuffling to a special factory that produced shrink-wrapped well-shuffled decks of cards. This factory would need a quality control department. It looks at the deck before shipping it to the customer, blocks some badly shuffled decks, and approves some others as well shuffled. But how is it possible if all $n!$ orderings of n cards have the same probability?

Here is a modernized version of the same paradox. Imagine that a company that runs a multiple-choice test for millions of students decided to make for each participant an individual version of the test by random permutation of possible answers to each question. Imagine that in one of the copies all the correct answers turn out to be labeled as "A". Should they discard this copy?

Current best practice

Whatever the philosophers say, statisticians have to perform their duties. Let us try to provide a description of their current best practice (see [**194, 175, 180**]).

A. *How a statistical hypothesis is applied.* First of all, we have to admit that probability theory makes no predictions but only gives recommendations: *If the probability* (computed on the basis of the statistical hypothesis) *of an event A is much smaller than the probability of an event B, then the possibility of the event B must be taken into consideration to a greater extent than the possibility of the event A* (assuming the consequences are equally grave). For example, if the

probability of A is smaller than the probability of being killed on the street by a meteorite, we usually ignore A completely (since we have to ignore event B anyway in our everyday life).

Borel [**22**, pp. 232–233] describes this principle as follows:

> ... Il y a à Paris moins d'un million d'hommes adultes; El's journaux rapportent chaque jour des accidents ou incidents bizarres arrivés à l'un d'eux; la vie serait impossible si chacun craignait continuellement pour lui-même toutes les aventures qu'on peut lire dans le faits divers cela revient à dire qu'on doit négliger pratiquement les probabilités inférieures à un millionième. ⟨...⟩
>
> *Souvent la peur d'un mal fait tomber dans un pire.*
>
> Pour savoir distinguer le pire, il est bon de connaître les probabilités des diverses éventualités....[1]

B. *How a statistical hypothesis is tested.* Here we cannot say naïvely that if we observe some event that has negligible probability according to our hypothesis, we reject this hypothesis. Indeed, this would mean that any 1000-bit sequence of the outcomes would make the fair coin assumption rejected (since this specific sequence has negligible probability 2^{-1000}).

Here algorithmic information theory comes into play: We reject the hypothesis if we observe a *simple* event that has negligible probability according to this hypothesis. For example, if coin tossing produces a thousand tails, this event is simple and has negligible probability, so we do not believe the coin is fair. Both conditions (simple and negligible probability) are important: the event "the first bit is a tail" is simple but has probability 1/2, so it does not discredit the coin. On the other hand, every sequence of outcomes has negligible probability 2^{-1000}, but if it is not simple, its appearance does not discredit the fair coin assumption.

Often both parts of this scheme are combined into a statement "events with small probabilities do not happen". For example, Borel writes: "...je suis arrivé à la conclusion qu'on ne devrait pas craindre d'employer le mot de *certitude* pour désigner une probabilité qui differe de l'unité d'une quantité suffisamment petite" ([**22**, p. 5]).[2] Sometimes this statement is called the "Cournot principle". But we prefer to distinguish between these two stages, because for the hypothesis testing the existence of a simple description of an event with negligible probability is important, and for application of the hypothesis it seems unimportant. (We can expect, however, that events interesting to us have simple descriptions because of their interest.)

Simple events and events specified in advance

Unfortunately, this scheme remains not very precise: the Kolmogorov complexity of an object x (defined as the minimal length of the program that produces x) depends on the choice of programming language. We need also to fix some way to

[1] Fewer than a million people live in Paris. Newspapers daily inform us about the strange events or accidents that happen to some of them. Our life would be impossible if we were afraid of all adventures we read about. So one can say that from a practical viewpoint we can ignore events with probability less that one millionth... *Often by trying to avoid something bad we are confronted with even worse...* To avoid this, it is good to know the probabilities of different events.

[2] I came to the conclusion that one must not be afraid to use the word *certainty* to describe a probability that falls short of unity by a sufficiently small quantity.

describe the events in question. Both choices lead only to an $O(1)$-change asymptotically; however, strictly speaking, due to this uncertainty we cannot say that one event has smaller complexity than the other one. (The word "negligible" is also not very precise.) On the other hand, the scheme described, while very vague, seems to be the best approximation to the current practice.

One of the possible ways to eliminate complexity in this picture is to say that a hypothesis is discredited if we observe a very improbable event *that was specified in advance* (before the experiment). Here we come to the following question. Imagine that you make some experiment and get a sequence of a thousand bits that looks random at first. Then somebody comes and says, "Look, if we consider every third bit in this sequence, the zeros and ones alternate." Will you still believe in the fair coin hypothesis? Probably not, even if you haven't thought about this event before while looking at the sequence: the event is so simple that one *could* think about it. In fact, one may consider the union of all simple events that have small probability, and it still has small probability (if the bound for the complexity of a simple event is small compared to the number of coin tossings involved, which is a reasonable condition anyway). And this union can be considered as specified before the experiment (e.g., it is described in this book).

On the other hand, if the sequence repeats some other sequence observed earlier, we probably will not believe it is obtained by coin tossing even if this earlier sequence had high complexity. One may explain this opinion saying the the entire sequence of observations is simple since it contains repetitions; however, the first observation may not be covered by any probabilistic assumption. This could be taken into account by considering the *conditional* complexity of the event (with respect to all information available before the experiment).

The conclusion is that we may remove one problematic requirement (being simple in some vague sense) and replace it by another problematic one (being specified before the observation). Borel comments on the situation [**21**, pp. 111–112]:

> Disons un mot de la réflexion de Bertrand relativement au triangle équilatéral que formeraient trois étoiles; elle se rattache à la question du nombre rond. Si l'on considère un nombre pris au hasard entre 1.000.000 et 2.000.000 la probabilité pour qu'il soit égal à 1.342.517 est égale à un millionième; la probabilité pour qu'il soit égal à 1.500.000 est aussi égale à un millionième. On considérera cependant volontiers cette dernière éventualité comme moins probable que la première; cela tient à ce qu'on ne se représente jamais *individuellement* un nombre tel que 1.542.317; on le regarde comme le *type* de nombres d'apparences analogues et si, en le transcrivant, on modifie un chiffre, on s'en aperçoit à peine et l'on ne distingue pas 1.324.519 de 1.324.517: le lecteur a besoin de faire un effort pour s'assurer que les quatre nombres écrits dans le lignes précédentes sont tous différents.
>
> Lorsque l'on a observé un nombre tel que le précédent comme évaluation d'un angle en dixiémes de secondes centésimales, on ne songe pas à se poser la question de savoir qu'elle était la probabilité pour que cet angle fût précisément égal a $13°42'51''{,}7$ car

on ne se serait jamais posé cette question précise avant d'avoir mesuré l'angle. Il faut bien que cet angle ait une valeur et, qu'elle que soit sa valeur à un dixième de seconde près, on pourrait, après l'avoir mesurée, dire que la probabilité *a priori*, pour que cette valeur soit précisément telle qu'elle est, est un dix-millionième, et que c'est là un fait bien extraordinaire. ⟨...⟩

La question est de savoir si l'on doit faire ces mêmes réserves dans le cas où l'on constate qu'un des angles du triangle formé par trois étoiles a une valeur *remarquable* et est, par exemple, égal à l'angle du triangle équilatéral ⟨...⟩ ou à un demi-angle droit ⟨...⟩ Voici ce que l'on peut dire à se sujet: on doit se défier beaucoup de la tendance que l'on a à regarder comme *remarquable* une circonstance que l'on n'avait pas précisée *avant l'expérience*, car le nombre des circonstances qui peuvent apparaître comme remarquables, à divers points de vue, est très considérable.[3]

Frequency approach

The most natural and common explanation of the notion of probability says that probability is the limit value of frequencies observed when the number of repetitions tends to infinity. (This approach was advocated as the only possible basis for probability theory by Richard von Mises.)

However, we cannot observe infinite sequences, so the actual application of this definition should somehow deal with finite number of repetitions. And for a finite number of repetitions our claim is not so strong: We do not guarantee that frequency of tails for a fair coin is *exactly* 1/2. We say only that it is *highly improbable* that it deviates significantly from 1/2. Since the words *highly improbable* need to be interpreted, this leads to some kind of logical circle that makes the frequency approach much less convincing; to get out of this logical circle we need some version of the Cournot principle.

---

[3]Let us comment on Bertrand's observation (about an equilateral triangle formed by three stars); it is related to the idea of a "round number". Consider a random integer between 1 000 000 and 2 000 000. The probability that it is equal to 1 342 517 is one over million; the probability that it is equal to 1 500 000, is also one over million. However, the second event is often considered as something less likely than the first one. This is because nobody considers individually a number like 1 542 317. It is considered as an example of some type of numbers, and if we change accidentally one digit when copying such a number, it is hardly noticeable: 1 324 519 looks very similar to 1 324 517. A special effort is needed to check that the four numbers mentioned above are different.

When a number like this appears as an angle measured in centesimal seconds, we do not ask ourselves what is the probability that this angle is exactly $13°42'51''{,}7$ because we never would be interested in such a question before the measurement. Of course, the angle should have some value, and whatever this value is (up to a tenth of a second), we may measure it and say that the *a priori* probability to get this value is one in ten million, so an extraordinary event has happened...

The question is whether the same reservations apply if one of the angles formed by three starts has a *remarkable* value, for example, is equal to the angle in the equilateral triangle... or the half of the right angle... What can we say about that? One should try hard to avoid the temptation to consider some event not fixed *before the experiment*, as a *remarkable* one, because a lot of events could look remarkable from some viewpoint.

Technically, the frequency approach can be related to the principles explained above. Indeed, the event "the number of tails in 1 000 000 coin tossings deviates from 500 000 more than by 100 000" has a simple description and very small probability, so we reject the fair coin assumption if such an event happens (and ignore the dangers related to this event if we accept the fair coin assumption). In this way the belief that frequency should be close to probability (if the statistical hypothesis is chosen correctly) can be treated as the consequence of the principles explained above.

Dynamical and statistical laws

We have described how probability theory is usually applied. But the fundamental question remains: Probability theory describes (to some extent) the behavior of a symmetric coin or die and turns out to be practically useful in many cases. But is it a new law of nature or some consequence of the known dynamical laws of classical mechanics? Can we somehow prove that a symmetric die indeed has the same probabilities for all faces (if the starting point is high enough and initial linear and rotation speeds are high enough)?

Since it is not clear what kind of "proof" we would like to have, let us put the question in a more practical way. Assume that we have a die that is not symmetric and we know exactly the position of its center of gravity. Can we use the laws of mechanics to find the probabilities of different outcomes?

It seems that this is possible, at least in principle. The laws of mechanics determine the behavior of a die (and therefore the outcome) if we know the initial point in the phase space (initial position and velocity) precisely. The phase space, therefore, is split into six parts that correspond to six outcomes. In this sense there is no uncertainty or probabilities up to now. But these six parts are well mixed since very small modifications affect the result, so if we consider a small (but not very small) part of the phase space around the initial conditions and any probability distribution on this part whose density does not change drastically, the measures of the six parts will follow the same proportion.

The last sentence can be transformed into a rigorous mathematical statement if we introduce specific assumptions about the size of the starting region in the phase space and variations of the density of the probability distribution on it. It then can be proved. Probably it is a rather difficult mathematical problem not yet solved, but at least theoretically the laws of mechanics allow us to compute the probabilities of different outcomes for a non-symmetric die.

Are "real-life" sequences complex?

The argument in the preceding section would not convince a philosophically minded person. Well, we can (in principle) compute some numbers that can be interpreted as probabilities of the outcomes for a die, and if we do not need to fix the distribution on the initial condition, it is enough to assume that this distribution is smooth enough. But still we speak about probability distributions that are somehow externally imposed in addition to dynamical laws.

Essentially the same question can be reformulated as follows. Make 10^6 coin tosses and try to compress the resulting sequence of zeros and ones by a standard compression program, say, `gzip`. (Technically, you need first to convert a bit sequence into a byte sequence.) Repeat this experiment (coin tossing plus gzipping)

as many times as you want, and this will never give you more than 1% compression. (Such a compression is possible for less than a 2^{-10000}-fraction of all sequences.) This statement deserves to be called a law of nature: it can be checked experimentally in the same way as other laws. So the question is, Does this law of nature follow from dynamical laws we know?

To see where the problem is, it is convenient to simplify the situation. Imagine for a while that we have discrete time, phase space is $[0,1)$, and the dynamical law is

$$x \mapsto T(x) = \textbf{ if } 2x < 1 \textbf{ then } 2x \textbf{ else } 2x - 1.$$

So we get a sequence of states $x_0, x_1 = T(x_0), x_2 = T(x_1), \ldots$; at each step we observe where the current state is—writing 0 if x_n is in $[0,1/2)$ and 1 if x_n is in $[1/2,1)$.

This transformation T has the mixing property we spoke about: If for some large t we look at the set of points that after t iterations are in the left half of the interval, we see that it is just the set of reals where tth bit of the binary representation is zero, and these reals occupy about a half in every (not too short) interval. In other words, we see that a sequence of bits obtained is just the binary representation of the initial condition. So our process just reveals the initial condition bit by bit, and any statement about the resulting bit sequence (e.g., its incompressibility) is just a statement about the initial condition.

So what? Do we need to add to the dynamical laws just one more metaphysical law saying that the world was created at a random (=incompressible) state? Indeed, algorithmic transformations (including dynamical laws) cannot increase significantly the Kolmogorov complexity of the state, so if objects of high complexity exist in the (otherwise deterministic, as we assume for now) real world now, they should be there at the very beginning. (Note that it is difficult to explain the randomness observed saying that we just observe the world at random time or in a random place. The number of bits needed to encode the time and place in the world is not enough to explain an incompressible string of length, say 10^6, if we use standard estimates for the size and age of the world. The logarithms of the ratios of the maximal and minimal lengths (or time intervals) that exist in nature are negligible compared to 10^6, and therefore the position in space-time cannot determine a string of this complexity.)

Should we conclude then that instead of playing dice (as Einstein could put it), God provided "concentrated randomness" (a state of high Kolmogorov complexity) while creating the world?

Randomness as ignorance: Blum–Micali–Yao pseudo-randomness

This discussion becomes too philosophical to continue it seriously. However, there are important mathematical results that could influence the opinion of the philosophers discussing the notions of probability and randomness if they knew these results. In this book we did not touch complexity with bounded resources (an important but not well-studied topic) and instead stayed in the realm of general computability theory, but we cannot avoid this topic when discussing the philosophical aspects of the notion of probability.

This result is the existence of pseudo-random number generators (as defined by Blum, Micali and Yao; they are standard tools in computational cryptography; see, e.g., the Goldreich textbook [**61**]). Their existence has been proven using

some complexity assumptions (the existence of one-way functions) that are widely believed though not yet proven.

Let us explain what a pseudo-random number generator (in the Blum–Micali–Yao sense) is. Here we use rather vague terms and oversimplify the matter, but there is rigorous mathematics behind it. Imagine a simple and fast algorithmic procedure that gets a *seed*, a binary string of moderate size, say, 1000 bits, and produces a very long sequence of bits out of it, say, of length 10^{10}. By necessity the output string has small complexity compared to its length (complexity is bounded by the seed size plus the length of the processing program, which we assume to be rather short). However, it may happen that the output sequences will be "indistinguishable" from truly random sequences of length 10^{10}, and in this case the transformation procedure is called a pseudo-random number generator.

It sounds like a contradiction: as we have said, output sequences have small Kolmogorov complexity, and this property distinguishes them from most of the sequences of length 10^{10}. So how they can be indistinguishable? The explanation is that the difference becomes obvious only when we know the seed used for producing the sequence, but there is no way to find out what seed is by looking at the sequence itself. The formal statement is quite technical, but its idea is simple: Consider any simple test that looks at a 10^{10}-bit string and says yes or no (by whatever reason; any simple and fast program could be a test). Then consider two ratios: (1) the fraction of bit strings of length 10^{10} that pass the test (among all bit strings of this length) and (2) the fraction of seeds that lead to a 10^{10}-bit string that passes the test (among all seeds). The pseudo-random number generator property guarantees that these two numbers are very close.

This implies that if some test rejects most of the pseudo-random strings (produced by the generator), then it would also reject most of the strings of the same length, so there is no way to find out whether somebody has given us random or pseudo-random strings.

In a more vague language, this example shows us that randomness may be in the eye of the beholder, i.e., the randomness of an observed sequence could be the consequence of our limited computational abilities which prevent us from discovering non-randomness. (However, if somebody shows us the seed, our eyes are immediately opened, and we see that the sequence has very small complexity.)

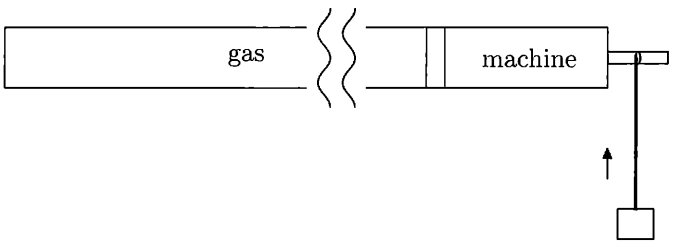
So we should not exclude the possibility that the world is governed by simple dynamical laws and its initial state can be also described by several thousands of bits. In this case "true" randomness does not exist in the world, and every sequence of 10^6 coin tossings that happened or will happen in the foreseeable future produces a string that has Kolmogorov complexity much smaller than its length. However, a computationally limited observer (like ourselves) would never discover this fact.

A digression: Thermodynamics

The connection between statistical and dynamical laws was discussed a lot in the context of thermodynamics while discussing the second law. However, one should be very careful with exact definition and statements. For example, it is often said that the second law of thermodynamics cannot be derived from dynamical laws because they are time-reversible while the second law is not. On the other hand, it is often said that the second law has many equivalent formulations, and one of them claims that the perpetual motion machine of the second kind is impossible,

i.e., no device can operate on a cycle to receive heat from a single reservoir and produce a net amount of work.

However, as Nikita Markaryan explained (personal communication), in this formulation the second law of thermodynamics *is* a consequence of dynamic laws. Here is a sketch of this argument. Imagine that a perpetual motion machine of the second kind exists. Assume this machine is attached to a long cylinder that contains warm gas. Fluctuations of gas pressure provide a heat exchange between gas and machine. On the other side the machine has a rotating spindle and a rope to lift some weight (due to rotation).



When the machine works, the gas temperature (energy) goes down and the weight goes up. This is not enough to call the machine a perpetual motion machine of the second kind (indeed, it can contain some amount of cold substance to cool the gas and some spring to lift the weight). So we assume that the rotation angle (and the height change) can be made arbitrarily large by increasing the amount of the gas and the length of the cylinder. We also need to specify the initial conditions of the gas; here the natural requirement is that the machine works (as described) for most initial conditions (according to the natural probability distribution in the gas phase space).

Why is such a machine impossible? The phase space of the entire system can be considered as a product of two components: the phase space of the machine itself and the phase space of the gas. The components interact, and the total energy is constant. Since the machine itself has some fixed number of components, the dimension of its component (or the number of degrees of freedom in the machine) is negligible compared to the dimension of the gas component (resp. the number of degrees of freedom in the gas). The phase space of the gas is split into layers corresponding to different levels of energy; the higher the energy is, the more volume in the phase space is used. This dependence outweighs the similar dependence for the machine since the gas has many more degrees of freedom. Since the transformation of the phase space of the entire system is measure-preserving, it is impossible that a trajectory started from a large set with high probability ends in a small set: the probability of this event does not exceed the ratio of a measures of destination and source sets in the phase space. So the machine that (with high probability) cools the gas in a random state and produces mechanical energy (=is a perpetual mobile of the second kind) is impossible.

This argument is quite informal and ignores many important points. For example, the measure on the phase space of the entire system is not exactly a product of measures on the gas and machine coordinates; the source set of the trajectory can have small measure if the initial state of the machine is fixed with very high precision, etc. (The latter case does not contradict the laws of thermodynamics: if

the machine uses a fixed amount of cooling substance of very low temperature, the amount of work produced can be very large.) But at least these informal arguments make plausible that dynamic laws make impossible the perpetual motion machine of the second kind (if the latter is defined properly).

Another digression: Quantum mechanics

Another physics topic often discussed is quantum mechanics as a source of randomness. There were many philosophical debates around quantum mechanics. However, it seems that the relation between quantum mechanical models and observations resembles the situation with probability theory and statistical mechanics. The difference is that in quantum mechanics the model assigns *amplitudes* (instead of probabilities) to different outcomes (or events). The amplitudes are complex numbers and the quantum Cournot principle says that if the (absolute value) of the amplitude of event A is smaller than for event B, then the possibility of event B must be taken into consideration to a greater extent than the possibility of event A (assuming the consequences are equally grave). Again this implies that we can (practically) ignore events with very small amplitudes.

The interpretation of the square of amplitude as probability can be then derived is the same way as in the case of the frequency approach. If a system is made of N independent identical systems with two outcomes 0 and 1 and the outcome 1 has amplitude z in each system, then for the entire system the amplitude of the event "the number of 1's among the outcomes deviates significantly from $N|z|^2$" is very small (it is just the classical law of large numbers in disguise).

One can then try to analyze measurement devices from the quantum mechanical viewpoint and prove (using the same quantum Cournot principle) that the frequency of some outcome of measurement is close to the square of the length of the projection of the initial state to a corresponding subspace outside some event of small amplitude, etc.

APPENDIX 2

Four algorithmic faces of randomness

V. USPENSKY

This appendix is a translation of the brochure "Four algorithmic faces of randomness" (2nd corrected edition, MCCME Publishers, Moscow, 2009; the first edition was published in 2006) that is based on a lecture delivered by Uspensky during the summer school "Modern Mathematics" (Dubna near Moscow, Russia, July 23, 2005). The terminology used in this brochure[1] is somewhat different from that used in the rest of the book; in particular, the terms *chaotic*, *typical*, and *unpredictable* are used to stress specific properties of random objects that appear in the corresponding definition. Chaoticness means that the complexity is high (no regularities can be used to give a short description); typicalness is based on measure theory; unpredictability guarantees that no strategy can win in a prediction game against this sequence. There are rigorous definitions for these notions that can be considered possible definitions of *true randomness*. And it is remarkable that natural definitions of chaoticness and typicalness turn out to be equivalent (Levin–Schnorr theorem).

Introduction

If somebody tells us that she tossed a "fair" coin twenty times and got the string

(I) 10001011101111010000

(where 0 and 1 denote head and tail), or the string

(II) 01111011001101110001,

this would not surprise us. However, if somebody claims to obtain

(III) 00000000000000000000

or

(IV) 01010101010101010101,

we start to doubt that the experiment was really performed in a proper way. But why?

[1]The same terminology was approved by Kolmogorov and used in the opening talk "Algorithms and randomness" at the First World Congress of the Bernoulli Society (written by Kolmogorov and Uspensky, delivered by Uspensky), and in [**84, 208, 194, 139**]

Somehow the strings (I) and (II) are perceived as "random" while (III) and (IV) are not.

But what does it mean to be "perceived as random"? Classical probability theory says nothing about this natural question. Sometimes they say that the outcomes (III) and (IV) have very small probability 2^{-20} to appear in a fair coin tossing, so the chances to get them are less than one in a million. Still, (I) and (II) have exactly the same probability!

Let us start with three important remarks.

- First, the intuitive idea of randomness depends on the assumed probability distribution. If the coin is very asymmetric and one side is much heavier, or if it is tossed in a very special way, (III) or (IV) may not surprise us. So, for simplicity, we will speak mostly about fair coin tossing, i.e., independent trials with success probability 1/2.
- Second, the intuitive idea of randomness has sense only if the string is long enough. It would be stupid to ask which of four strings 00, 01, 10, 11 looks more random than the others.
- Finally, there is no sharp boundary between (intuitively) random and non-random strings. Indeed, changing one bit in a random string, we get a string that is random, too. But in several steps we can obtain (III) or (IV) from any string. This well-known effect is sometimes called "heap paradox".

So, trying to define randomness, one should consider very long strings, or, even better, infinite bit sequences (in general infinite objects are "approximations from above" for large finite objects). For infinite sequences one may try to draw a meaningful sharp division between random and non-random objects, i.e., to define rigorously a mathematical notion of a ***random bit sequence***. In this survey we describe several attempts to provide such a definition, made by different authors. However, a general disclaimer is needed: for all practical purposes only finite sequences (strings) matter, so these definitions are necessarily far from "real life". In fact, even very long finite sequences never appear in real life, so it is hard to extend our intuition of randomness even to long finite strings. This said, we now switch to mathematical definitions.

Let us start with some useful notation and terminology.

We consider finite *bit strings*, i.e., finite sequences of zeros and ones. (They are also called *binary words*.) A string $x = x_1, \ldots, x_n$ has length n, denoted also by $|x|$.[2] A string may have zero length, i.e., may contain no bits; it is then called an *empty string* and is denoted by Λ.

The set of all binary strings is denoted by Ξ. The set of all infinite bit sequences is denoted by Ω. An infinite sequence $a_1, a_2, a_3, \ldots$ has finite string $a_1, a_2, \ldots, a_n$ as its *n-bit prefix*. For every string x we consider the set $\Omega_x \subset \Omega$ of all infinite sequences that have prefix x. This set is called a *ball*, and the *volume* of this ball is defined as $2^{-|x|}$ and denoted by $\mathbf{v}(x)$.[3]

Each sequence from Ω is considered as a record of an (infinite) coin tossing. Let us repeat that for now we assume that the coin is fair. Mathematically speaking, it

[2]We used the notation $l(x)$ for the length of x in the main part of the book.

[3]In the rest of the book we call Ω_x an interval, not a ball, and speak about its length, not volume.

means that we consider a *uniform probability distribution* on Ω where for each ball Ω_x the probability to get an element of Ω_x is equal to its volume.

Our goal is to specify a well-defined subset of Ω that could be considered as the set of all random sequences. Traditional probability theory cannot help here; even the question can hardly be stated in its language. In a paradoxical way, the notion of algorithm helps. It may sound strange: the notion of randomness is defined in terms of the notion of algorithm, which is a deterministic procedure that has nothing to do with randomness, but it is the case. All known definitions of randomness for individual objects (in our case—individual binary sequences) are based on the theory of algorithms in some way.

We may start by trying to identify a characteristic property that intuitively should be possessed by all random sequences, and then use this property (specified rigorously) as a formal definition of randomness.

So, what properties could be reasonably expected from a randomly chosen bit sequence?

First of all, the limit frequency should exist in such a sequence. For the simplest case of a fair coin this means that the fraction of zeros (as well as the fraction of ones) in the n-bit prefix of the sequence should converge to $1/2$ as n goes to infinity. This property can be called *frequency stability*. Moreover, the same property should hold not only for the sequence itself, but also for every its *reasonably chosen* subsequence.

Second, a randomly chosen sequence is expected to be *chaotic*. This means that it has a complex structure and cannot have a *reasonable* description. The psychological difference between the perception of strings (I), (II) and (III), (IV) can be explained, as Kolmogorov suggested, by the fact that strings (I) and (II) have no short description while (III) and (IV) have a regular structure and can be described easily.

Third, a randomly chosen sequence should be *typical*, in the sense that it belongs to any *reasonable* majority.

Finally, it should be *unpredictable*. This means that making bets against this sequence, trying to guess its terms, we cannot win systematically, and no clever strategy could help us.

Of course, these wordings are vague. One should specify the meaning of word "reasonable" that occurs in the explanations of frequency stability, chaoticness, and typicalness, as well of the words "description" and "strategy". Theory of algorithms can be used to convert these descriptions into formal definitions, and we get four rigorously defined properties: *frequency stability*, *chaoticness*, *typicalness*, and *unpredictability*. Each of them can be considered as some "algorithmic face of randomness" and can to some extent pretend to be a mathematical definition of randomness. In this way we get four well-defined classes of sequences that could compete for the title of the "true class of random sequences" though each has its strong and weak points.

In the following exposition our goal is two-fold: (1) to give rigorous definitions for the four properties mentioned above and therefore to define four classes of sequences; (2) to state (currently known) relations between these properties (and, therefore, between the corresponding classes of sequences).

Face one: Frequency stability and stochasticness

The idea to define the notion of an individual random sequence goes back to Richard von Mises, a well-known German mathematician; it seems that he was the first who tried to give such a definition. This happened in the beginning of the twentieth century, in 1919. At least it was he who suggested a reasonable approach to this definition (though he did not give a rigorous mathematical one).

Von Mises started by requiring frequency stability, i.e., the existence of limit frequency: the fraction of ones among the first n terms should converge (for the case of fair coin) to $1/2$ as n tends to infinity. Of course, this property is not sufficient. For example, this is true for the (definitely non-random) sequence

$$0, 1, 0, 1, 0, 1, 0, 1, \ldots .$$

Evidently, we should require that not only the sequence itself, but also its subsequences satisfy the frequency stability property. But we cannot expect *all* the subsequences to be stable in this sense: indeed, even a perfectly random sequence has a zero subsequence, we may select just the terms that are equal to zero. So we have to restrict ourselves and consider only "reasonable chosen", or "admissible" subsequences.

It is nice to consider any subsequence of a given sequence as the result of selection procedure applied to the terms of the original sequence: the subsequence consists just of those terms which are selected. Any selection procedure is based on some selection rule. To obtain a reasonable, or admissible, subsequence, one needs to use a reasonable (admissible) selection rule. For example, a reasonable selection rule may select all terms a_i where i is a prime number, or all terms that follow zeros (i.e., all terms a_{i+1} such that $a_i = 0$). In this way we get two admissible subsequences.

Kolmogorov at some point suggested the name *stochastic* for a sequence whose admissible subsequences all have the frequency stability property.

The scheme suggested by von Mises was rather vague; it was turned to a rigorous definition of randomness when the theory of algorithms was developed. One of its inventors, an American mathematician Alonzo Church suggested in 1940 to define the admissible selection rule as algorithms of special type. The sequences where all Church-admissible subsequences satisfy the frequency stability property are called *Church stochastic* sequences.[4] This definition, however, looks too broad: for example, there exists a Church stochastic sequence that becomes non-Church-stochastic after a computable permutation of its terms.[5]

In 1963 Kolmogorov modified the definition given by Church and suggested a broader class of admissible selection rules, thus defining a broader (in fact, strictly broader) class of admissible subsequences. In particular, Kolmogorov's definition does not require that the selected terms keep the ordering they had in the original sequence. A corresponding class of sequences, called *Kolmogorov stochastic sequences*,[6] appears: they are sequences such that all Kolmogorov-admissible subsequences satisfy the frequency stability property. By definition, this class is a

---

[4] In the main part of the book they are called *Mises–Church random* sequences.

[5] See Theorem 203(d), p. 307.

[6] They are called *Mises–Kolmogorov random sequences* in the main part of the book. The most standard name used nowadays is *Kolmogorov–Loveland stochastic sequences*.

subclass (in fact, a proper subclass) of the class of Church stochastic sequences. In the sequel we denote the class of stochastic sequences by **S**.

Soon it turned out that the class **S** was also too broad. For example, one may construct a Kolmogorov stochastic sequence where each prefix has more zeros than ones.[7] It contradicts our intuition (supported by some theorems of probability theory: a one-dimensional random walk returns to the starting point with probability 1). So even the strictest version of the von Mises approach currently known does not provide an intuitively satisfactory notion of randomness, though it is an interesting object to study that reflects some aspects of randomness.

To be precise, let us reproduce the definitions suggested by Church and Kolmogorov. In both cases we define some class of *admissible selection rules* used to form subsequences of a given sequence.

Imagine that the terms of the sequence (zeros and ones) are written on paper cards that are put on the table, face down, so we do not see what is written on the cards. Our goal is to select some of the cards and form another sequence made of the bits on the selected cards. This subsequence (in the case of Kolmogorov's definition this term is used in a broad sense, the order of terms in the subsequence may differ from their order in the original sequence) is called an admissible subsequence. An admissible selection rule is an algorithm that decides on each step (1) which bit should be revealed (corresponding card turned over) next and (2) whether this bit should be included in the subsequence or not. The algorithm has access to the bits already revealed (those bits form its input). It may well happen that the algorithm selects only finitely many bits (it may hang or reveal more and more bits without selecting any of them), in this case we say that no admissible subsequence is formed. (Anyway, the frequency stability property makes sense only for infinite sequences.) If for every admissible selection rule we get a sequence that satisfies the frequency stability property, the original sequence is called stochastic.

To give a more precise description, let us recall some terminology. A function is called *computable* if there is an algorithm that *computes* this function. This means, for some function f, that (1) the algorithm terminates on every input x such that $f(x)$ is defined, and produces $f(x)$, and (2) the algorithm does not terminate on all inputs where f is undefined.

Assume that a sequence $a_1, a_2, \ldots$ is given, so the nth card contains bit a_n. A *Church admissible selection rule* is an arbitrary computable function G defined on all binary strings and has *True* and *False* as values. The cards are turned over sequentially (first the card that carries a_1, then a_2, etc.); before the next card is turned over, the selection rule decides whether that card is selected or not. This is done in the following way. Assume that n cards, carrying bits $a_1, \ldots, a_n$, have been turned over. If $G(a_1, \ldots, a_n)$ equals *True*, then the next card, carrying a_{n+1}, is included in the subsequence; otherwise, it is not. At the first step we include a_1 in the subsequence depending on the value of $G(\Lambda)$. In other words, the selected subsequence consists of terms

$$a_{n(1)}, a_{n(2)}, a_{n(3)}, \ldots,$$

where $n(1), n(2), n(3), \ldots$ are all numbers n such that $G(a_1, \ldots, a_{n-1}) = \mathit{True}$, assuming that there are infinitely many numbers with this property. Otherwise, we get a finite sequence, and it is not considered as admissible subsequence.

[7] See Theorem 203(b), p. 307.

This was Church's definition. Before we explain Kolmorogov's version, let us explain what we mean by a *generalized subsequence* of some sequence $a_1, a_2, \ldots$. It is a sequence of the form

$$a_{\varphi(1)}, a_{\varphi(2)}, \ldots, a_{\varphi(k)}, \ldots,$$

where

$$i < j \Rightarrow \varphi(i) \neq \varphi(j).$$

In the usual definition of subsequence the last condition is stronger: we require that subsequence is monotone, i.e., $\varphi(i) < \varphi(j)$ for $i < j$.

Each *Kolmogorov admissible selection rule* attempts to select some generalized subsequence of the given sequence. Here we say "attempts" since this attempt may be unsuccessful: in this case instead of an infinite subsequence we get a tuple (finite sequence) that consists of some terms taken from the original sequence. We say that our original sequence is *Kolmogorov stochastic* if all infinite subsequences obtained from it by Kolmogorov admissible rules have the frequency stability property.

It remains to explain what is a Kolmogorov admissible selection rule. To specify such a rule, we consider two computable functions F and G. The first one (F) is used to construct some intermediate generalized subsequence; the final subsequence is a (monotone) subsequence of that intermediate sequence. Both functions F and G are defined on (some) binary strings, so their domains are subsets of Ξ (may be, different ones). The values of F are positive integers, and the values of G are Boolean values *True* and *False*. We start by constructing a sequence of natural numbers

$$n(1) = F(\Lambda),\ n(2) = F(a_{n(1)}),\ \ldots,\ n(k+1) = F(a_{n(1)}, \ldots, a_{n(k)}).$$

This construction is stopped and gives a finite sequence in the following three cases:

- the value $F(a_{n(1)}, \ldots, a_{n(k)})$ is undefined;
- the value $G(a_{n(1)}, \ldots, a_{n(k)})$ is undefined;
- the value $F(a_{n(1)}, \ldots, a_{n(k)})$ coincides with one of the $n(1), \ldots, n(k)$.

If none of these three events happens, we get an infinite sequence of indices

$$n(1), n(2), n(3), \ldots,$$

and a generalized subsequence $a_{n(1)}, a_{n(2)}, a_{n(3)}, \ldots$. Now, and this is the last step, we select a (monotone) subsequence of these subsequence by choosing all terms $a_{n(k)}$ such that $G(a_{n(1)}, \ldots, a_{n(k-1)}) = \mathit{True}$, in the order of increasing k.

Face two: Chaoticness

Let us return to strings (I)–(IV) that we started with. According to Kolmogorov's explanation, strings (I) and (II) look random because they are *complex*, while (III) and (IV) look non-random because they are *simple*. It seems that intuitively we expect the result of a random process be complex, and we suspect some cheating when it turns out to be simple.

There are many ways to compare objects around us: we can distinguish big and small objects, or heavy and light objects. Also we can speak about complex and simple objects. In the 1960s Kolmogorov[8] observed that mathematics can be used

[8]Kolmogorov's paper of 1965 [**78**] became most well known, but he was not alone: many people independently came to similar ideas. As Kolmogorov notes in his paper [**79**], the first publication in this direction was written by Ray Solomonoff [**187**]; Gregory Chaitin [**28**] also developed this idea a bit later.

for such a classification. Now the corresponding mathematical theory is usually called *Kolmogorov complexity theory.*

The main idea is simple and natural: ***complexity of an object can be measured by the length of its shortest description.*** Each object has a long description, however a complex object cannot have a short description.

Let Y be the set of all objects we consider, and let X be a set of all possible descriptions of those objects. Let us recall that $|x|$ stands for the length of x. According to what we said, the complexity of an object y, denoted by $\mathrm{Comp}(y)$, is defined by the formula

$$\mathrm{Comp}(y) = \min_x\{|x|\colon x \text{ is a description of } y\}.$$

If an object y has no description at all, its complexity is infinite (the minimum of the empty set is defined as infinity).

Of course, we need some uniform way to measure the length of a description, it would be unfair to say that something can be easily described in Chinese because only one glyph is needed, and has only a complicated English description that consists of several dozen letters. So we assume that all descriptions are presented as binary strings. In other words, we assume in the sequel that $X = \Xi$.

The set of all pairs $\langle x, y\rangle$ where x describes y, can be called a *language of descriptions* or a *description language.* Note that (for some description language) some object y may have many descriptions. We may also consider description languages where the same x can describe several objects. For example, the expression "a string of zeros" can be considered as a description of all such strings, and we may even consider an expression "a bit string" as a description of all binary strings.[9]

What has been discussed above was a preparation for the following formal definition. Consider an arbitrary subset E in the Cartesian product $\Xi \times Y$, called a *description language.* If $\langle x, y\rangle \in E$, we say that the string x is a *description* of the object y. The *complexity* Comp_E of an object y with respect to the description language E is defined as

$$\mathrm{Comp}_E(y) = \min_x\{|x|\colon \langle x, y\rangle \in E\}.$$

(Again, the minimum of the empty set is infinite.)

For a language $E = \Xi \times Y$ where every string x is a description of every object y, the complexity of all objects equals zero, since the empty string is a description of every object. Such a description language is formally allowed but will not appear in the classes of description languages considered in the sequel.

Imagine two description languages with the following property: to get a description of some object y for the second language, we take its description for the first language and repeat it twice. Evidently the second description language is worse, since it provides descriptions that are twice as long, and we want the descriptions to be short.

Formally speaking, we say that a description language A is *not worse* than a description language B and write $A \leqslant B$, if there exists some constant c such that $\mathrm{Comp}_A(y) < \mathrm{Comp}_B(y) + c$ for all y.

Consider natural languages as description languages. Assume that for any pair of natural languages there is a translation algorithm that converts any given text

[9]However, we should not go too far in this direction; otherwise, the notion of complexity will be trivial.

in the first language into an equivalent text in the second language. We then can conclude that description language corresponding to the second language is not worse than that corresponding to the first language. For example, a Turkish-language description of an object may consist of two parts: a Japanese-language description and a Japanese–Turkish translation algorithm. In this way we get a Turkish description that is longer than a Japanese description at most by a constant (the length of the Japanese–Turkish translation algorithm). This constant does not depend on the choice of the object described. Taking the shortest possible Japanese description, we conclude that the Turkish language is not worse than the Japanese language if we consider both as description languages.

Let us call a *language family* any family of description languages. Having some language family $\mathcal{L}$, we may ask whether there exists an optimal language in this family. A language A from $\mathcal{L}$ is *optimal* (for $\mathcal{L}$) if it is not worse than any other description language in the family, i.e., if

$$(\forall B \in \mathcal{L})\ (A \leqslant B).$$

An optimal description language, if it exists for some family, should be used to measure complexity. The complexity of an object with respect to some fixed optimal description language can be called *algorithmic entropy* of this object.[10] Entropy is the final version of the measure of complexity (when some family of description languages is fixed).

For some language families one can prove the existence of an optimal description language. For those families the notion of entropy is well defined. The statements of this type are usually called *Solomonoff–Kolmogorov theorems*, since they were first to discover such statements.

A given family may contain (and usually contains) many optimal description languages. Each of them gives some entropy function. However, due to the optimality definition, every two entropies (corresponding to two optimal description languages for some family) differ by at most an additive constant. In other words, if A and B are two optimal description languages in the family $\mathcal{L}$, then there exists a constant c such that

$$|\mathrm{Comp}_A(y) - \mathrm{Comp}_B(y)| < c$$

for all y.

REMARK. Of course, one can rightfully complain that the notion of entropy that pretends to be a complexity measure for individual objects is still defined only up to some bounded additive term, and one would like to select some *true* entropy function among different ones. However, attempts of this type have not succeeded up to now.

We use the letter K to denote algorithmic entropy (as a tribute to Kolmogorov)[11] and sometimes add another letter to specify the family of description languages used. If K' and K'' are two entropy functions for the same family of description languages, then

$$|K' - K''| < c$$

(as we have noted).

---

[10] In the main part of the book we keep the name *complexity* for this notion, and we use the word *entropy* for Shannon entropy only.

[11] In the main part of the book the letter K is used for prefix version of complexity (entropy).

Kolmogorov not only gave a definition of algorithmic entropy, but also realized its connection with randomness. He observed that for a random sequence the entropy of its n-bit prefix grows fast as n tends to infinity. Notice that a random sequence can start with, say, a million zeros, and the entropy of this prefix is very low, but asymptotically it still grows fast.

When speaking about prefixes of binary sequences, we use binary strings (such as (I), (II), (III), (IV)) as objects whose complexity is measured. So we assume that $Y = \Xi$ in the sequel.

If a description language contains a pair $\langle z, z\rangle$, this means that z is its own description. Consider a description language D that consists of all such pairs; this D can be called a *diagonal* language (as mathematicians would say); linguists could call it an *antonymous* description language. Evidently, $\mathrm{Comp}_D(y) = |y|$. Let us consider only language families that include D (the family of monotone description languages defined in the sequel, has this property). Then for every entropy function K for this family there exists some c such that

$$K(y) < |y| + c$$

for all y. So, up to an additive constant, the maximal possible value of entropy for an n-bit string is n. Kolmogorov conjectured that for a random sequence this upper bound for its n-bit prefixes is tight (again up to a constant). This is how Kolmogorov interpreted the chaoticness property.

So let us fix some language family (that contains an optimal language), and let K be one of the corresponding entropy functions. A sequence

$$a_1, a_2, \ldots, a_n, \ldots$$

is then called *chaotic* if there exists a constant c such that

$$K(a_1, a_2, \ldots, a_n) > n - c$$

for all n. Evidently, this definition does not depend on the choice of specific entropy function in the family, but may depend on the choice of the family.

It turned out that for some natural language family the notion of chaoticness defined in this way gives a reasonable formalization of the intuitive idea of randomness.

In Kolmogorov complexity theory the relations between descriptions and objects have an algorithmic nature. Following Kolmogorov, we restrict ourselves to *enumerable*[12] sets. The notion of an *enumerable set* is one of the main notions in the theory of computability (and in mathematics in general). It can be explained intuitively in the following way. Imagine a printing device that prints binary strings sequentially; printed strings are separated by spaces. The time intervals between printing consecutive strings may be arbitrary (but each string should be printed completely without delays, and infinite sequences of bits are not allowed). It may happen that the device hangs (and does not print anything) after finitely many strings have been printed, then the set of strings printed by the device is finite. In particular, the device may print nothing at all, then we get an empty set of output strings. For such a device, the set of all printed strings is enumerable—and every enumerable set can be obtained in this way, if the device is equipped with a

[12]What we call *enumerable* is usually called *computably enumerable*, or *recursively enumerable*. The word enumerable usually refers to countable sets. In our exposition, we use the term *enumerable sets* to refer to computably enumerable sets; see footnote 13 below.

suitable program. For example, for every formal theory (like set theory, or formal arithmetic) the set of all theorems (provable statements) is enumerable. The introduction of a formal computational model or of a general notion of a formal theory falls beyond our scope. However we will describe the notion of an ***enumerable set*** in more detail.

Let us start with countable sets. This term is used in two different ways. One more narrow definition says that countable sets are those sets for which there exists a one-to-one correspondence with the set $\mathbb{N}$ of all natural numbers. The other more liberal definition says that countable sets are those sets for which there exists a one-to-one correspondence with some initial segment of $\mathbb{N}$. Here by *initial segment* of $\mathbb{N}$ we mean a subset M of $\mathbb{N}$ that is downward-closed, i.e., every natural number that is smaller than some element of M also belongs to M. For example, the entire $\mathbb{N}$ and the empty set $\varnothing$ are both initial segments of $\mathbb{N}$, and all finite sets are countable in this more liberal interpretation. We use this interpretation; then one can say that *a set is countable if it is either empty or can be represented as a set of terms of an infinite sequence.* For example, the finite set $\{a, b, c\}$ is the set of terms of infinite sequence $a, b, c, c, c, c, \ldots$. If we additionally require that this infinite sequence is computable, we get the definition of an enumerable set. It remains to explain what a computable sequence is.

A sequence $w_1, w_2, \ldots, w_n, \ldots$ is called *computable* if there exists an algorithm that for any given n computes its nth term w_n. One may say that the notion of a computable sequence is an effective (algorithmic) version of the notion of sequence, and the notion of an enumerable set is an effective (algorithmic) version of the notion of a countable set.[13] Let us repeat the definition: *a set is enumerable if it is empty or it is a set of terms of some computable sequence.*

All the description languages we consider are subsets of $\Xi \times \Xi$ and therefore are all countable. Kolmogorov suggested considering enumerable description languages only. The final step in the definition of chaoticness was made by Leonid Levin, a student of Kolmogorov; in 1973 he published a paper in which a class of monotone description languages was introduced, and the corresponding notion of chaoticness was studied.[14] Let us provide the corresponding definitions.

We say that strings u and v are *compatible* and write $u \approx v$ if one of these strings is a prefix of the other one.

A description language E is called *monotone* if E is enumerable and the following requirement is satisfied:

$$\big(\langle x_1, y_1\rangle \in E \ \&\ \langle x_2, y_2\rangle \in E \ \&\ (x_1 \approx x_2)\big) \Rightarrow (y_1 \approx y_2).$$

It can be shown that there exists a monotone description language that is optimal for the family of monotone description languages. So the notion of entropy for this family is well defined; the corresponding entropy function is called the *monotone entropy*[15] and is denoted by KM.

---

[13] To stress the difference between algorithmic and non-algorithmic notions, enumerable sets are usually called *recursively enumerable* or *computably enumerable* (computable functions were traditionally called "recursive functions" for historical reasons). The word "enumerable" is often used as a synonym for "countable".

[14] A similar notion was introduced by Claus-Peter Schnorr in his publication of 1972; see the footnote on p. 482.

[15] In the main part of the book this function is called *monotone complexity*; it is defined in Section 6.2.

A sequence that is chaotic for monotone description languages is called just *chaotic* in the sequel.[16] The chaoticness requirement can be written as follows:

$$\exists c\ \forall n\ (KM(a_1, a_2, \ldots, a_n) > n - c).$$

We denote the class of all chaotic sequences by **C**.

It seems that the definition of chaoticness is a good approximation to the intuitive notion of randomness. There are two reasons for this.

First, every chaotic sequences satisfies the standard laws of probability theory (such as the strong law of large numbers, the law of iterated logarithm etc.).

Second, the class **C** of chaotic sequences coincides with another natural candidate for the randomness definition, the class **T** of typical sequences (see below):

$$\mathbf{C} = \mathbf{T}.$$

One could even use the names *typical-chaotic* or *chaotic-typical* for the sequences in **C** (=**T**) and denote this class by **CT** or **TC**. This class is a proper subclass of the class **S** of all Kolmogorov stochastic sequences (as we said, the definition of stochasticity seems to be too liberal to reflect our intuition of randomness):

$$\mathbf{TC} \subset \mathbf{S}, \quad \mathbf{TC} \neq \mathbf{S}.$$

Face three: Typicalness

What do we mean by saying that some object is "typical" for some category? This means that it belongs to every reasonable majority of objects selected from this category. For example, a typical human being has height less than 2 meters (i.e., belongs to the majority of people who have height less than 2 meters), has age at least 3 (i.e., belongs to the majority of people who are at least 3 years old), etc. The adjective "reasonable" is important here, since every object x is doomed to fall outside the overwhelming majority of objects that differ from x.

Our intuition says that every random object is typical. But how can we clarify the latter notion? Let us give a mathematical definition of typicalness for a bit sequence (assuming the uniform distribution on infinite bit sequences that corresponds to a fair coin tossing). As we have said, for that we need to specify what an "overwhelming majority" is in the set of all sequences and when that majority is "reasonable". Then the class of typical sequences is defined as the intersection of all reasonable overwhelming majorities.

A set of sequences forms an overwhelming majority if its complement is small, so we need to define the notion of a small set. Using the language of probability theory, we can say that some set Q is small if the event "randomly chosen sequence is in Q" has probability zero. In terms of measure theory small sets are just sets of measure 0. However, we want to have a more explicit definition. It can be given in the following way.

A set Q is *small* if it can be covered by a countable family of balls whose total volume is arbitrarily small. In other terms, Q is small if for every natural m there exists a sequence of binary strings

$$\langle x(1), x(2), \ldots, x(n), \ldots \rangle$$

[16] Since this property is equivalent to Martin-Löf randomness (called typicalness in this appendix), we do not use a different name in the main text of the book.

such that

$$Q \subset \bigcup_n \Omega_{x(n)},$$

$$\sum_n \mathbf{v}(x(n)) = \sum_n 2^{-|x(n)|} < \frac{1}{m}.$$

Evidently, each sequence forms a small set (a singleton), so the intersection of all sets with small complements is empty, and we need to define "reasonable overwhelming majority" in a more restrictive way.

This can be done by considering the following effective version of the definition of a small set.

First, we require the sequence $\langle x(1), x(2), \ldots, x(n), \ldots\rangle$ in the definition to be computable. In other words, some algorithm should compute $x(n)$ given n as input.

Second, we require not only the computability of this sequence, but *uniform* computability: the sequence $\langle x(1), x(2), \ldots, x(n), \ldots\rangle$ with required properties can be constructed ***by some algorithm*** given m. We need to explain what it means: this sequence is an infinite object, and algorithms deal with finite objects only. We require that there exists some algorithm that, given m, produces an algorithm (=a program) that computes some sequence $\langle x(1), x(2), \ldots, x(n), \ldots\rangle$ with required properties.[17]

These two changes in the definition of a small set give us a definition of a more restricted notion, that of an *effectively small set.*[18] The complements of effectively small sets could be called *effectively large* sets. Now the intersection of all effectively large sets is not empty; moreover, this intersection itself is an effectively large set. This smallest effectively large set is our goal: we denote it by **T** and call it the set of all *typical* sequences.

Typical sequences are usually called *Martin-Löf random* sequences, since this definition was suggested (as a definition of randomness) in 1966 by Per Martin-Löf, an eminent Swedish mathematician, who in 1964 and 1965 studied at Moscow University under the supervision of Kolmogorov.

As we have said already, the class **T** of all typical sequences coincides with the class **C** of all chaotic sequences,

$$\mathbf{T} = \mathbf{C},$$

and the elements of this class can be called *chaotic-typical* or *typical-chaotic* sequences (and the class may be denoted by **CT** or **TC**).

As we have already mentioned,

$$\mathbf{CT} \subset \mathbf{S}, \quad \mathbf{CT} \neq \mathbf{S}.$$

Face four: Unpredictability

Any random sequence is unpredictable in the following sense: if we know the values of some its terms, it does not give us any information about the terms not revealed yet. So if a Casino prepares a random sequence and then allows a Player to make bets on the values of the terms she does not know, the Casino is safe; more precisely, there is no strategy for the Player that allows her to make Casino bankrupt independent of the initial amount of money Casino has.

---

[17] An equivalent definition requires that, given m and n, an algorithm computes the nth term of a sequence that satisfies the requirements for the given m.

[18] In the main part of the book those sets are called *effectively null* sets.

In other words, we define the unpredictability of some sequence in terms of a game where Casino uses that sequence and Player makes bets against that sequence, i.e., on the values of terms of that sequence not yet revealed. Player and Casino initially have some amount of money. Casino also has some bit sequence, and Player does not know it. Player can then make bets about some bits of that sequence, not necessarily in the monotone order and not necessarily about all bits; some terms of the sequence may be skipped.

We can imagine that bits are written on cards that lie on an infinite table face down, so Player does not see the bits: she sees only an infinite sequence of card backs. At each move, Player points to some card, makes a prediction about the bit on that card and declares the amount of her bet. Then the card is turned over. If the prediction is correct, Casino pays that amount to Player; if the prediction is wrong, Player loses her money (i.e., pays that amount to Casino). Player wins if she managed to make Casino bankrupt. Of course, if Player has unlimited credit resources, she can always win by doubling the bets until her guess becomes correct. But we assume that Player has no credit line, so the amount of the bet should not exceed her current capital.

A sequence is called *predictable* if there is a strategy for Player that allows her to win against that sequence. This means that for the arbitrarily large initial capital of Casino, Casino will nevertheless become bankrupt if Player uses this strategy. A sequence is called *unpredictable* if it is not predictable.

More formally the game may be described as follows. We consider an infinite sequence of zeros and ones:

$$\mathbf{a} = \langle a_1, a_2, a_3, \ldots \rangle.$$

At each move Player creates a triple

$$\langle n, i, v \rangle,$$

where

$$n \in \mathbb{N}, \quad i \in \{0, 1\}, \quad v \in \mathbb{Q}, \quad v \geqslant 0;$$

here, as usual, $\mathbb{N}$ is the set of natural numbers,[19] and $\mathbb{Q}$ is the set of rational numbers. The meaning of this triple is the following: n is the number of the bit on which the bet is made, i is the predicted value of that bit, and the non-negative rational number v is the amount of the bet. The moves are performed sequentially, starting from the first one; the triple that represents the kth move is denoted by $\langle n(k), i(k), v(k) \rangle$. (More formally, moves *are* triples of the described form.)

Player's capital before the kth move is denoted by $V(k-1)$. Without loss of generality we may assume that the initial capital of Player equals 1, i.e., $V(0) = 1$.

After each move, Player's capital changes according to the following rules:

- if $i(k) = a_{n(k)}$ (Player made a correct guess), then $V(k) = V(k-1) + v(k)$;
- if $i(k) \neq a_{n(k)}$ (Player made an incorrect guess), then $V(k) = V(k-1) - v(k)$.

Two additional remarks are needed.

[19] Sometimes 0 is considered as a natural number (logicians and computability experts usually do this), sometimes not—in this appendix we follow the second convention and do not consider 0 as a natural number.

First, moves may be *valid* or *invalid*, and the game continues only if the move is valid. By definition, a valid move should satisfy two requirements:

1) the number of the bit on which bet is made is *valid*: this means that this bit was not used earlier, i.e., $n(k)$ does not appear among $n(1), \ldots, n(k-1)$;
2) the bet itself is *valid*: its size is less than the current capital, i.e., $v(k) < V(k-1)$.

The game stops when Player makes an incorrect move. In this case she keeps the current capital forever, and cannot win.

It is also possible that Player refrains from making any move (she may even refrain from making the first move); in this case she also keeps the current capital forever and cannot win. However, we do not say in this case that the game is stopped. Player can think for an arbitrarily long time before making her next move; the time for thinking is not limited, so it is possible that she thinks forever, i.e., never makes any move. While thinking, the capital remains unchanged, so in this case the capital remains unchanged forever. We do not say, however, that the game is stopped, since Player never explicitly declares that she will not make any move. So three scenarios are possible: (1) Player makes infinitely many moves; (2) Player attempts to make an invalid move and the game is stopped; (3) Player at some point starts thinking but never makes a move.

Of course, this is only an illustration, and the formal definition goes as follows. By definition, Player wins against the sequence $\mathbf{a}$ if

$$\sup_k V(k) = +\infty,$$

i.e.,

$$\forall W \ \exists k \ V(k) > W.$$

This means that Player can cause the bankruptcy of Casino independently of its initial capital. This is possible only if game is infinite, that is, at each turn Player makes a valid move.

The game is described now, and we define the notion of *strategy*. A strategy is a rule that tells Player what she should do, i.e., prescribes the next move based on the history of the game. The strategy is not required to be total, its output may be undefined because Player makes no move: the strategy produces an output exactly in the cases when Player makes some move. The input to the strategy is the history of the game, that is, the sequence of all the moves made so far and the values of the bits revealed so far. (One could add to the history the information about the capital at every moment, but this is redundant, since this information can be easily computed.)

Here is the history before the kth move can be represented as a table:

$$\begin{array}{cccc} n(1) & n(2) & \ldots & n(k-1) \\ i(1) & i(2) & \ldots & i(k-1) \\ v(1) & v(2) & \ldots & v(k-1) \\ a_{n(1)} & a_{n(2)} & \ldots & a_{n(k-1)} \end{array}$$

(for $k = 1$ the table is empty).

A strategy therefore is a function that maps every table of this kind to a move $\langle n, i, v \rangle$, or it may be undefined (on some tables). Here "table of this kind" means an

arbitrary table with positive integers in the first row, non-negative rational numbers in the third row, and bits in the second and fourth rows.

Assume that we are given a strategy and a table that can appear during the game of that strategy against some sequence. Then the first three rows of that table can be uniquely reconstructed from the last row. Indeed, we reconstruct the first move $\langle n(1), i(1), v(1)\rangle$ applying the strategy to the empty table. Then (assuming that the fourth row is known) we know the history of the game before the second move, i.e., the table

$$\begin{array}{c} n(1) \\ i(1) \\ v(1) \\ a_{n(1)}. \end{array}$$

Then we apply the strategy again to find the second move $\langle n(2), i(2), v(2)\rangle$ and hence the table

$$\begin{array}{cc} n(1) & n(2) \\ i(1) & i(2) \\ v(1) & v(2) \\ a_{n(1)} & a_{n(2)}, \end{array}$$

and so on.

So, when defining strategies, we may assume that only the fourth line of the table is given to the strategy. This line is a binary string (an element of Ξ). Given a binary string, the strategy may have no output or provide the next move, an element of $\mathbb{N} \times \{0,1\} \times \mathbb{Q}_+$, as an output. (Here $\mathbb{Q}_+$ stands for the set of all non-negative rational numbers.)

So we can now give the final definition of a strategy: it is a partial mapping of type

$$\Xi \to \mathbb{N} \times \{0,1\} \times \mathbb{Q}_+.$$

We are interested in strategies that are computable, i.e., that can be computed by an algorithm. Let us specify what that means. Assume that an algorithm $\mathbf{A}$ gets elements of a set X as input and produces elements of a set Y as output. Consider the subset of X that consists of all inputs for which $\mathbf{A}$ provides some output, and the function from this subset of X to Y that maps each input value to the corresponding output value. We say that $\mathbf{A}$ *computes* that function, and a function is *computable* if some algorithm computes it.

We will consider strategies that are computable in this sense. (If the algorithm does not terminate for an input history, then the strategy is undefined on that history, in which case we may imagine that Player is thinking about her move but never comes to any decision.)

We say that a sequence $\mathbf{a}$ is *predictable* if there exists a computable strategy that wins against $\mathbf{a}$ (i.e., Player wins if she uses this strategy against $\mathbf{a}$). Otherwise, $\mathbf{a}$ is *unpredictable.*[20] The class of all unpredictable sequences is denoted by $\mathbf{U}$.

It is known that every unpredictable sequence is Kolmogorov stochastic (it belongs to the class $\mathbf{S}$) and that every typical-chaotic sequence is unpredictable:

$$\mathbf{CT} \subset \mathbf{U} \subset \mathbf{S}.$$

[20] In the main text unpredictable sequences are called "Kolmogorov–Loveland random"; see the discussion on p. 310.

It is also known that the class of Kolmogorov stochastic sequences is significantly larger than the class of unpredictable sequences:

$$\mathbf{S} \neq \mathbf{U}.$$

But the question whether the classes of chaotic (=typical) and unpredictable sequences coincide, is still open:

$$\mathbf{CT} \stackrel{?}{=} \mathbf{U}.$$

This is an important problem; several people tried to solve it but got only partial results.

Strategies that avoid invalid moves. Defining unpredictable sequences, we may restrict ourselves to strategies that never make invalid moves. Indeed, we can modify an algorithm **A** that computes the winning strategy, and get another algorithm **B** that does not terminate when **A** attempts to make an invalid move. One has to check whether the move is valid, and this can be done algorithmically: knowing the input for **A**, we reconstruct the history of the game, including the numbers of bits revealed and the current capital of Player, so we can check the validity of the move recommended by **A** and cancel an invalid attempt.[21]

Generalization for arbitrary computable distributions

Up to now we considered only the case of *uniform distribution* on the space Ω of binary sequences; all the main ideas can be illustrated in this special case. Now, to complete the picture, we consider the general case of arbitrary *computable probability distribution* on Ω. (See the definition below.) Let us make some comments for readers who are not yet familiar with the general notion of a *probability distribution* (*measure*).

We say that a set M is equipped with a *measure* μ if (1) some class of subsets of M is chosen and its elements are called *measurable subsets*; (2) for each measurable subset A some number $\mu(A)$ is chosen and this number is called the *measure* of A. There are some requirements (axioms of measure theory); we do not go into detail here and note only that this requirement implies the following fact: any finite or countable union of disjoint measurable subsets is measurable and its measure is equal to the sum of the measures of the parts. For *probability measures*, or *probability distributions*, we require also that $\mu(M) = 1$. The intuitive meaning of $\mu(A)$ is the probability of the event "a randomly chosen element of M belongs to A".

A measure on Ω is determined by the measures of balls. For the uniform distribution (and only for it) we have

$$(\forall x \in \Xi)\ (\mu(\Omega_x) = 2^{-|x|}).$$

It corresponds to the case where zeros and ones are equiprobable and trials are independent. A slightly more general case is *Bernoulli distribution*, also called a *binomial distribution*. Here the trials are also independent, but in each trial the probabilities of 1 and 0 are p and $1-p$, respectively. This number p is a parameter;

[21] A more complicated argument shows that the class of unpredictable sequences does not change if we consider only total computable strategies, i.e., the strategies defined on all inputs; see the discussion on p. 310.

for $p = 1/2$, we get uniform distribution. Formally, for the Bernoulli distribution with parameter p we have

$$\mu(\Omega_x) = p^k(1-p)^{|x|-k},$$

where k is the number of ones in x.

The next step is to consider *quasi-Bernoulli distributions* where trials are still independent, but the probability of success may depend on the number of the trial: in the kth trial the outcome 1 appears with probability $p(k)$. More formally, consider a sequence of reals

$$\mathbf{p} = \langle p(1), p(2), \ldots, p(k), \ldots \rangle, \quad 0 \leqslant p(k) \leqslant 1.$$

Then the *quasi-Bernoulli distribution with parameter* $\mathbf{p}$ is defined by the formula

$$\mu(\Omega_x) = \prod_{i=1}^{n} r_i,$$

where $r_i = p(i)$ if $x_i = 1$ and $r_i = 1 - p(i)$ if $x_i = 0$. If $\mathbf{p} = \langle p, p, \ldots, p, \ldots \rangle$, we get Bernoulli distributions as a special case.

In this section we show how the definitions of stochasticness, chaoticness, typicalness, and unpredictability can be extended to the case of arbitrary computable probability distribution μ (see the definition below). Let us tell in advance that for this more general case the same relationships hold:

$$\mathbf{C}(\mu) = \mathbf{T}(\mu) \subset \mathbf{U}(\mu) \subset \mathbf{S}(\mu),$$
$$\mathbf{S}(\mu) \neq \mathbf{U}(\mu)$$

(the last inequality is true assuming that all balls have positive measure).

Here $\mathbf{C}(\mu)$, $\mathbf{T}(\mu)$, $\mathbf{U}(\mu)$, $\mathbf{S}(\mu)$ denote (respectively) the classes of chaotic, typical, unpredictable, and Kolmogorov stochastic sequences with respect to the distribution μ; these classes are defined below. Our old classes now can be written as $\mathbf{C} = \mathbf{C}(\eta)$, $\mathbf{T} = \mathbf{T}(\eta)$, $\mathbf{U} = \mathbf{U}(\eta)$, and $\mathbf{S} = \mathbf{S}(\eta)$ for the uniform distribution η on Ω.

Let us warn the reader that this section is addressed to people who like generalizations. It is a bit more difficult than the previous exposition. Moreover, our task, that is the search for a natural definition of randomness, is less clear for general distributions. The intuitive meaning of an individual random sequence as a plausible outcome of some natural physical process like coin tossing becomes less and less clear as we switch from the simple example of fair coin tossing and the uniform distribution to more and more general classes of distributions.

Computable measures (distributions). One may attempt to call a measure on Ω computable if there exists an algorithm that for each binary string x computes a measure $\mu(\Omega_x)$ of the ball Ω_x. However, we have to be cautious: the output of an algorithm may be an integer or rational number (to be more precise, its name or representation as a string over a finite alphabet), and we cannot name all the real numbers since we have only countably many names. So we require that the algorithm computes not a real number (the measure of the ball) but its approximation.

Here is the definition. A measure μ is *computable* if there exists an algorithm that for any given pair (a binary string x, a positive rational ε) computes a rational number that differs from $\mu(\Omega_x)$ at most by ε.

One could add a requirement that there is an algorithm that for a given x says whether the equality $\mu(\Omega_x) = 0$ holds or not. That requirement gives a strictly smaller class of measures that are called *strongly computable* measures in the sequel.

An important subclass of the class of computable measures is the class of *computable-rational* measures where the measure of each ball Ω is a rational number that can be computed (the corresponding fraction presented) given x. Note that it is not the same as a computable measure whose values (on balls) are rational numbers: in the latter case we are only able to provide arbitrarily close approximations to the rational number which is the measure of the ball, and this is not enough to produce this number entirely (as a fraction of two integers).

Recalling that probability distributions are those measures for which the measure of Ω equals 1, we may speak about computable probability distributions on Ω. Many definitions and statements about randomness for the uniform distribution can be generalized naturally to arbitrary computable probability distributions. In particular, one can prove a general version of Martin-Löf's theorem (saying that the intersection of all effectively large sets is an effectively large set itself), and Levin's theorem (saying that typicalness is equivalent to chaoticness defined using monotone complexity).[22]

Stochasticness. Recall our notation: the kth term of some sequence $\mathbf{e}$ is denoted by e_k or (to avoid subscripts) by $e(k)$.

For the case of uniform distribution, stochasticness was understood as *global frequency stability*, i.e., the stability of frequencies *in all admissible subsequences.* Those subsequences were obtained by application of Kolmogorov-admissible selection rules. For the general case of an arbitrary computable measure this scheme remains the same, but frequency stability should be replaced by some more general property derived from the strong law of large numbers in probability theory.

For the Bernoulli distribution with parameter $p \in (0, 1)$ the definition is clear: we require that every admissible subsequence has the frequency stability property with limit frequency p. In other words, for every admissible subsequence the fraction of ones in its n-bit prefixes tends to p as $n \to \infty$. We also treat the cases $p = 0$ and $p = 1$ in a special way: only the sequence that contains only zeros (respectively, ones) is stochastic.

Can we consider an even more general case of non-Bernoulli distribution? This definitely goes beyond the original idea of von Mises: he tries to define the notion of probability as limit frequency in random sequences. Still one can try to follow this path, starting with quasi-Bernoulli sequences.

One could not expect the existence of limit frequency in the subsequences of a quasi-Bernoulli sequence (and different subsequences may have different limit frequencies even if they exist). So the stochasticity requirement should take into account the selection rules (which terms were selected). But first let us exclude the case when a bit appears that has probability zero: we declare a sequence $\mathbf{a}$ non-stochastic if there exists some k such that $a(k) = 0$ and $p(k) = 1$, or $a(k) = 1$ and $p(k) = 0$. Assuming this does not happen, we call a sequence $\mathbf{a}$ *stochastic with*

[22] In the main part of the book this result is called the "Levin–Schnorr theorem"; Schnorr's paper was published earlier and considered some special notion of complexity called "process complexity". It can differ significantly from monotone complexity (see the section about history below and the bibliography at the end of this appendix), but the underlying ideas are similar and the proof for one of them can be easily adapted for the other one.

respect to a selection rule Θ if its generalized subsequence

$$\mathbf{b} = \langle a(m_1), a(m_2), \ldots, a(m_k), \ldots \rangle,$$

obtained by this rule, satisfies the following requirement taken from the strong law of large numbers for quasi-Bernoulli distributions:

$$\frac{a(m_1) + \cdots + a(m_k)}{k} - \frac{p(m_1) + \cdots + p(m_k)}{k} \to 0$$

as $k \to \infty$. Now we can define a stochastic sequence with respect to a given quasi-Bernoulli measure by requiring that for every Kolmogorov-admissible selection rule that produces an infinite generalized subsequence this subsequence is stochastic with respect to this rule.

REMARK. By definition, generalized subsequences are always infinite, so the word "infinite" in the last sentence can be omitted. However, we use it to stress that we really are interested only in the infinite sequences, not tuples.

Now we want to extend the notion of Kolmogorov-stochasticness to a wider class of probability distribution. First, let us introduce some notation.

Let $n(1), n(2), \ldots, n(k)$ be some natural numbers, and let $i(1), i(2), \ldots, i(k)$ be some bits. By $A^{n(1),\ldots,n(k)}_{i(1),\ldots,i(k)}$ we denote the set of all sequences $\mathbf{a} \in \Omega$ such that

$$(*) \qquad a_{n(1)} = i(1),\ a_{n(2)} = i(2),\ \ldots,\ a_{n(k)} = i(k).$$

The ratio

$$\frac{\mu\left(A^{n(1),\ldots,n(k),m}_{i(1),\ldots,i(k),1}\right)}{\mu\left(A^{n(1),\ldots,n(k)}_{i(1),\ldots,i(k)}\right)}$$

is denoted in the sequel by

$$\mu\left(\begin{array}{c|c} m & n(1), \ldots, n(k) \\ 1 & i(1), \ldots, i(k) \end{array}\right),$$

since it is the conditional probability of the event "the mth term of $\mathbf{a}$ equals 1" under condition $(*)$. That probability is undefined when the denominator equals zero.

Let us fix an arbitrarily sequence $\mathbf{a} \in \Omega$ and some Kolmogorov-admissible selection rule Θ. Our goal is to define the notion "$\mathbf{a}$ is stochastic with respect to Θ". Recall that Θ was applied to select a subsequence of $\mathbf{a}$ in two steps. First, we select an auxiliary generalized subsequence $\mathbf{c}$; then the resulting subsequence is obtained by omitting some terms in $\mathbf{c}$. More precisely,

$$\mathbf{c} = \langle a_{n(1)}, a_{n(2)}, \ldots, a_{n(k)}, \ldots \rangle,$$

where the number $n(k)$ is computed algorithmically given the tuple

$$\langle a_{n(1)}, a_{n(2)}, \ldots, a_{n(k-1)} \rangle.$$

Then, using the same tuple as input, the rule Θ decides whether the term $a_{n(k)}$ should be included in the final subsequence $\mathbf{b}$. Therefore,

$$\mathbf{b} = \langle a(n(k_1)), a(n(k_2)), \ldots, a(n(k_j)), \ldots \rangle.$$

At both stages it may happen that Θ (the corresponding algorithm) does not produce any output (the number in the first case, and the decision bit in the second case). Then $\mathbf{b}$ is finite and we do not require anything, hence the sequence $\mathbf{a}$ is

declared to be stochastic with respect to Θ. But if **b** is infinite, then some requirement should be fulfilled to make **a** stochastic with respect to Θ. Let us describe that requirement.

By r_j we denote the conditional probability

$$\mu\left(\begin{pmatrix} n(k_j) \\ 1 \end{pmatrix} \middle| \begin{matrix} n(1), & n(2), & \ldots, & n(k_j-1) \\ a(n(1)), & a(n(2)), & \ldots, & a(n(k_j-1)) \end{matrix}\right).$$

Consider the difference

$$\delta_j = \frac{r_1 + r_2 + \cdots + r_j}{j} - \frac{a(n(k_1)) + a(n(k_2)) + \cdots + a(n(k_j))}{j}.$$

Here δ_j is defined only if all $r_1, \ldots, r_j$ are defined.

We say that **b** *satisfies the strong law of large numbers* if all δ_j are defined and $\delta_j \to 0$ as $j \to \infty$. A sequence **a** is then called *stochastic with respect to* Θ if the generalized subsequence obtained from **a** according to Θ satisfies the strong law of large numbers.

Finally, a sequence **a** is called *Kolmogorov stochastic* with respect to a given probability distribution if **a** is stochastic with respect to every Kolmogorov-admissible selection rule that selects an infinite generalized subsequence from **a**.

This definition by itself does not use the computability of the measure. However, to compare it with other randomness notions, we need to assume that the measure (the probability distribution in question) is computable.

Chaoticness. A sequence $\mathbf{a} = \langle a_1, a_2, a_3, \ldots \rangle$ is chaotic with respect to a computable measure μ if there exists a constant c such that

$$KM(a_1, a_2, \ldots, a_n) > -\log \mu(\Omega_{a_1, a_2, \ldots, a_n}) - c,$$

for all n (here log stands, as usual, for the binary logarithm).

For arbitrary computable measures, the motivation for this definition is the same as it was for the uniform measure. One can prove that for every computable measure μ there exists some c such that

$$KM(x) < -\log \mu(\Omega_x) + c$$

for all strings x. Informally speaking, for every computable measure μ we can find some monotone description language that fits that measure in the following sense: it provides short descriptions for strings x that have big values of $\mu(\Omega_x)$ (as the inequality above specifies). The sequence is chaotic if those descriptions cannot be significantly shortened (more than by a constant).

Typicalness. The definition of typicalness can be naturally extended to arbitrary measures: we used the volume (=the uniform measure) of balls when defining small sets, and now we should use their measure instead.

As before, we start by defining effectively small sets. A set $Q \subset \Omega$ is *effectively small with respect to measure* μ if there exists an algorithm **A** with the following property. Given any positive integer m and input, the algorithm **A** produces as output an algorithm for the computing a sequence $\langle x(1), x(2), \ldots, x(n), \ldots \rangle$ such

that

$$Q \subset \bigcup_n \Omega_{x(n)},$$

$$\sum_n \mu(\Omega_{x(n)}) < \frac{1}{m}.$$

Then a set is considered *effectively large with respect to* μ if its complement is effectively small.

For every computable measure μ the following *Martin-Löf theorem* holds: the union of all effectively small sets is effectively small, and therefore the intersection of all effectively large sets is effectively large. This result provides the smallest effectively large set which is called the *constructive support of measure* μ. The elements of this constructive support are called *typical with respect to* μ, so the set $\mathbf{T}(\mu)$ is defined as the constructive support of the distribution μ.

Unpredictability. Let us explain how the definition of unpredictability (given above for the uniform distribution) should be changed for the case of arbitrary distributions. Two changes are necessary for that: some auxiliary factor (that equals 1 for the uniform distribution and was therefore omitted), and some additional rule that tells us when to stop the game (for the uniform distribution it is not needed since the corresponding situation cannot happen).

The payoff for bets depends on the probability distribution. If Player makes a wrong guess, her bet is lost, i.e., the capital decreases by the size of the bet. But if she makes a correct guess, the increase is proportional to the bet, and the coefficient depends on the probability of the correctly predicted outcome. The coefficient is large if this outcome has small probability, and is small if it has large probability. For uniform distribution the probability is always $1/2$ and the coefficient is always 1. The exact value of the coefficient for an arbitrary distribution is determined as follows.

Recall that a_k denotes the kth term of a sequence $\mathbf{a}$; similarly, a'_k is the kth term of $\mathbf{a}'$, etc. Player's jth move is a triple $\langle n(j), i(j), v(j)\rangle$.

Let $\mathbf{a}$ be the sequence used by Casino for the game. Let

$$A(k-1) = \{\mathbf{a}' \in \Omega\colon a'_{n(j)} = a_{n(j)} \text{ for all } j = 1, 2 \ldots, k-1\}$$

(so $A(0) = \Omega$) and

$$A_i(k) = \{\mathbf{a}' \in A(k-1)\colon a'_{n(k)} = i\} \text{ for } i = 0, 1.$$

This notation makes sense if all the numbers $n(l)$ appearing in it are defined. Note that

$$\Omega = A(0) \supset A(1) \supset A(2) \supset \cdots, \tag{1}$$

$$1 = \mu(A(0)) \geqslant \mu(A(1)) \geqslant \mu(A(2)) \geqslant \cdots. \tag{2}$$

If Player's kth guess was correct, then

$$i(k) = a_{n(k)}, \quad A_{i(k)}(k) = A(k); \tag{3}$$

otherwise

$$i(k) \neq a_{n(k)}, \quad A_{1-i(k)}(k) = A(k). \tag{4}$$

Note also that

$$A(k-1) = A_0(k) \cup A_1(k). \tag{5}$$

If $i(k) = a_{n(k)}$ (i.e., the kth guess was correct), Player's capital increases according to the formula

$$V(k) = V(k-1) + v(k) \cdot \frac{\mu(A_{1-i(k)}(k))}{\mu(A_{i(k)}(k))}. \tag{6}$$

This formula guarantees that the game is fair, i.e., the expected change of the capital at the kth step equals zero. However, an unpleasant surprise is possible when we apply this rule: the value $\mu(A_{i(k)}(k))$ in the denominator may be equal to 0. In this case (which was not possible for the uniform distribution nor for any *positive* distribution where all balls have positive measures) a special additional stopping rule is used.

Additional stopping rule. This is used when it happens (*for the first time*) that $\mu(A(k)) = 0$ (cf. equation (2) above). Assume that $\mu(A(k-1)) \neq 0$, $\mu(A(k)) = 0$. The last move made was the kth move, when Player made a prediction $i(k)$. If the prediction turns out to be correct (i.e., $i(k) = a_{n(k)}$), then the game is stopped and Player's capital is declared to be infinite $V(k) = +\infty$, and Player wins the game. If the prediction turns to be incorrect, i.e., $i(k) \neq a_{n(k)}$, then the game is also stopped, but in this case the capital of Player remains unchanged (and fixed), so Player does not win the game.

This rule takes care of the problem of a zero denominator in (6). Indeed, (6) is applied only if $i(k) = a_{n(k)}$. In this case $A_{i(k)}(k) = A(k)$, according to (3). So if we get a zero denominator, it means that $\mu(A(k)) = 0$. But in this case we apply the additional stopping rule instead of (6). (Or we could say that we apply (6) and declare that we get $+\infty$ when dividing positive number $\mu(A_{1-i(k)}(k))$ by zero.)

The definitions of a strategy, a computable strategy, a strategy that performs only valid moves remain (up to these changes) the same as for the special case of the uniform distribution.

History and bibliography

We print the numbers in italic to distinguish them from the references in the main list of references.

[*1*] A. Kolmogorov, V. Uspensky. "Algorithms and randomness." *SIAM J. Theory Probab. Appl.*, v. 32 (198), p. 389–412. Translated with annoying errors (for instance, everywhere instead of the correct translation "recursively enumerable" an incorrect translation "countable" is used); better translation can be found in: Yu. V. Prokhorov, V. V. Sazonov, eds., *Proc. 1st World Congress of the Bernoulli Society (Tashkent 1986)*, v. 1, *Probability Theory and Appl.*, VNU Science Press, Utrecht, 1987, p. 3–55.

[*2*] V. Uspensky, A. Semenov. *Algorithms: main ideas and applications*. Kluwer Academic Publishers, 1993, 269 pp.

[*3*] V. Uspensky, A. Semenov, A. Shen. "Can an individual sequence of zeros and ones be random?" *Russian Math. Surveys*, v. 45(1), 1990, 121–189.

[*4*] V. Uspensky, A. Shen. "Relations between varieties of Kolmogorov complexities." *Mathematical Systems Theory*, v. 29 (3), 1996, p. 271–292.

[*5*] An. Muchnik, A. Semenov, V. Uspensky. "Mathematical metaphysics of randomness," *Theoretical Computer Science*, v. 207, 1998, p. 263–317.

[*6*] A. Shen, "On relations between different algorithmic definitions of randomness," *Soviet Math. Dokl.*, v. 38 (2), 1989, 316–319.

[7] V. V'yugin, "Algorithmic entropy (complexity) of finite objects and its application to defining randomness and amount of information," *Selecta Mathematica* (formerly *Sovietica*), v. 13(4), 1994, p. 357–389.

[8] M. Li, P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications.*, Springer-Verlag, 1993, xx+546 pp., 38 illustrations; third ed., 2008, xxiii+790 pp., 50 illustrations.

Of course, this list is not complete in any sense. However, in these publications (especially in [8]) one can find further references to get a more complete picture. In [2], in Section 2.6 (Applications to probability theory) different definitions of random sequence are given (pp. 166–178). Note that the terminology in [2] is different; what we call chaotic sequences is called there Kolmogorov random sequences, what we call typical sequences is called there Martin-Löf random sequences; Church stochastic sequences are called there Mises–Church random sequences, and Kolmogorov stochastic sequences are called there Mises–Kolmogorov–Loveland random sequences, and in [3] they are called Kolmogorov–Loveland stochastic sequences. D. Loveland independently discovered this class later, in 1966, while Kolmogorov's paper appeared in 1963. The unpredictable sequences (as defined by us) do not appear in [2] since they were introduced only later (in 1998, see [5]).

An example of a Church stochastic sequence that becomes not Church stochastic after a computable permutation of its terms was published by D. Loveland in 1966. That example is important not only because it shows a flaw in Church's definition, but also because it stresses an important property of randomness that is intuitively obvious but was not taken into account earlier: conservation after every computable permutation.

Description complexity theory, i.e., the theory of complexity of objects, should not be mixed with *computational complexity theory*, i.e., the theory of complexity of computations. Description complexity theory forms the basis for algorithmic information theory. Both theories, closely related, were founded by Kolmogorov in his seminar talks at Lomonosov Moscow State University in the beginning of the 1960s; Kolmogorov's main goal was to create a new foundation for information theory based on the idea that the more complex an object is, the greater is the information carried by that object. That new foundation should avoid the notion of probability replacing it by the notion of algorithm, and also should be applied to the definition of an individual random object. In his 1969 paper (the English version was published in 1968) Kolmogorov wrote:

> (1) Basic information theory concepts must and can be founded without recourse to probability theory, and in such a manner that "entropy" and "mutual information" concepts are applicable to individual values.
> (2) Thus introduced, information theory concepts can form the basis of the term *random*, which naturally suggests that randomness is the absence of regularities.[23]

The idea of measuring the complexity of an object by the length of its shortest description was proposed by Kolmogorov in his paper of 1965;[24] a year earlier similar ideas were published in the U.S. by Ray Solomonoff (Kolmogorov learned

[23]The published English version of this paper says "random is the absence of periodicity", but this is evidently a translation error, and we correct the text following the Russian version.

[24]See item [**78**] in the main list of references.

about Solomonoff's work when publishing his 1969 paper,[25] and he cited it). So we called the statement about existence of an optimal description language the *Solomonoff–Kolmogorov theorem.* At the same time (the middle of 1960s) Kolmogorov suggested in his seminar talks that the growth of complexity of prefixes can be used to define randomness for individual infinite sequences. However, the family of description languages introduced by Kolmogorov turned out to be unsuitable for this, and (as we have said before) a suitable family was found in 1973 by Leonid Levin who defined the notion of monotone entropy.

Typical sequences were defined (and called "random") by Per Martin-Löf in 1966, as we have said earlier.

The existence of a Kolmogorov stochastic sequence that is not typical (= not chaotic) was proven by Alexander Shen (see [6] or [2, Section 6.2.4]).[26]

Let K be one of the entropy functions (many of them were studied, including plain, a priori, monotone, process, prefix, and decision entropies; the versions mentioned are different in the sense that the difference between any two of these entropy functions is not bounded). We may try to define chaotic sequences (with respect to the uniform distribution) using K by requiring that

$$\exists c\ \forall n\ (K(a_1, a_2, a_3, \ldots, a_n) > n - c).$$

(Just for the record: for plain and decision entropy no sequences with this property exist, and for four other versions we get a definition that is equivalent to typicalness.) The equivalence of chaoticness for monotone entropy and typicalness was shown by Levin in the same paper where monotone entropy was introduced. Independently Claus-Peter Schnorr in his 1973 paper[27] (the conference version was published in 1972) introduced another version of entropy, *process entropy* (Schnorr used the name "process complexity") and proved (by a similar argument) that the corresponding notion of chaoticness is equivalent to typicalness. Process entropy and monotone entropy differ significantly (their difference is unbounded, as Vladimir Vyugin showed in [7]); later Schnorr switched to monotone entropy, and the equivalence between chaoticness based on monotone entropy and typicalness is sometimes called the *Levin–Schnorr theorem.*

Prefix entropy was introduced by Levin in his Ph.D. thesis submitted in 1971, but the thesis was rejected[28] and the definition was published only in 1974.[29] Later Gregory J. Chaitin independently discovered the same definition (see his paper "A theory of program size formally identical to information theory", *Journal of the Association of Computing Machinery*, 1975, v. 22, no. 3, 329–340) where he also introduced chaoticness definition using prefix entropy and claimed (without proof) that this version of chaoticness is equivalent to typicalness; the proof was first published in Vyugin's paper [7, Corollary 3.2]. Prefix entropy can be defined as

---

[25] See item **[79]** in the main list of references.

[26] The main idea of this proof was invented by M. van Lambalgen for monotone selection rules and can be easily generalized to non-monotone ones. — *A. Shen.*

[27] See item **[169]** in the main list of references.

[28] Levin was a USSR citizen. The rejection of his thesis, having been approved by Kolmogorov who was the thesis advisor and all the reviewers, took place for political reasons. He emigrated in 1978 and earned a Ph.D. at the Massachusetts Institute of Technology (MIT) in 1979.

[29] See item **[94]** in the main list of references, where the prefix entropy was called *prefix complexity*; we use the same name in the main part of this book.

entropy for the family of prefix description languages. A set E is a *prefix description language* if E is enumerable and the following condition holds:

$$\big(\langle x_1, y_1\rangle \in E \ \&\ \langle x_2, y_2\rangle \in E \ \&\ (x_1 \approx x_2)\big) \Rightarrow (y_1 = y_2).$$

Note also that the term "complexity" is normally used for what we call "entropy" (i.e., complexity with respect to an optimal description language).

Unpredictable sequences (as defined above) appeared (spring 1991) in the joint talk "Randomness and Lawlessness" given by Andrei Muchnik, Alexey Semenov, and Vladimir Uspensky at the conference in California devoted to the foundations of randomness (March 4–7, Institute for Mathematical Studies in the Social Sciences, Stanford University). The paper [5] published in 1998 is based on that talk, and it contained the results about relations between unpredictability and other randomness notions.[30]

Note that the definition of unpredictability given in the present exposition slightly differs from the definition in [5]. Namely, in [5] the bet was called valid if a weaker inequality $v(k) \leqslant V(k-1)$ holds, while we require the strict inequality $v(k) < V(k-1)$. Both definitions are equivalent (i.e., lead to the same class of unpredictable sequences), but still our current definition looks somehow more thoughtful. There are two reasons to prefer the new version. First, the game looks more natural: if Player bets all her capital and makes a wrong guess, then no money is left and the rest of the game is trivial (only zero bets are possible). Second, we need strict inequality to make the game realistic from the algorithmic viewpoint for arbitrarily computable measures (only computable-rational measures were considered in [5]). Indeed, before a bet is made, Player should check that the bet is valid. She can check the strict inequality $v(k) < V(k-1)$ before making the bet (checking algorithm terminates and confirms the inequality if it holds, and does not terminate otherwise), but one cannot construct a similar algorithm for the inequality $v(k) \leqslant V(k-1)$ and the arbitrary computable measure.

The game approach to randomness was mentioned already by von Mises who spoke about the non-existence of a winning strategy (without formal definitions) when playing against Casino. Later several formal definitions were suggested, but the version from [5] (with a cosmetic change mentioned above) seems to be more adequate. Indeed, in the previous versions either the computability requirement for the strategy was replaced by a requirement of another kind (still of an algorithmic nature, but less natural) or the resulting class of sequences was known to be different from the class of chaotic-typical sequences. For the definition from [5] there is still some hope that it is equivalent to chaoticness and typicalness; if it is indeed the case, this equivalence will be another reason to believe that this class (of chaotic-typical sequences) is a good approximation for our intuitive notion of a random sequence.

[30] For the case when the bets are made from left to right, as the sequence terms appears, the game approach to randomness and the corresponding notion of a martingale was introduced in the 1930s by Jean Ville [**206**] as an alternative to von Mises' approach.