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).

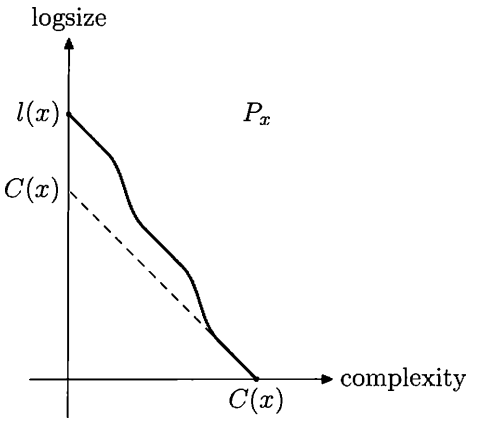

FIGURE 52. The set P_x

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.

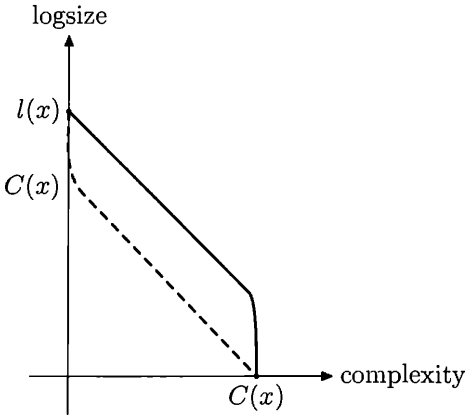

FIGURE 53. Two opposite possibilities for a boundary curve

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

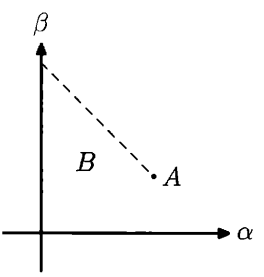

FIGURE 54. The parameters of the hypothesis A and its simplification B

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.