

Review of  
**An introduction to Kolmogorov Complexity and its Applications**  
**Second Edition, 1997**  
**Authors: Ming Li and Paul Vitanyi**  
**Publisher: Springer (Graduate Text Series)**

Reviewer: William Gasarch

## 1 Overview

The string 111111111111 looks “less random” than the string 100110001001. Kolmogorov complexity makes this intuitive notion of randomness rigorous. Once this is done, new questions arise and some old questions can be answered. This book spends half of its time making these notions rigorous, and the other half applying them. More precisely:

1. Chapters 1-4 carefully establish the rigorous definitions needed to study randomness. If this were the only goal it would not need four chapters; however, the authors also explore many issues that lead to the definitions and that are consequences of the definitions.
2. Chapters 5-7 apply Kolmogorov complexity to computer science; chapter 8 applies it to Physics. Most of the applications only use a small part of what is in chapter 1-4. This is good— if a reader is only interested in applications they can read these chapters having just learned a few things from chapters 1-4.

## 2 Summary of Contents

Chapter 1 contains motivation for the subject and a quick review of combinatorics, probability, and computability. The treatment is a good refresher course, but is too mature for a first time learner.

Chapter 2 defines the complexity of a string and some notions of randomness. Informally, the complexity of a string  $x$  is the shortest description of  $x$ . We give a formal definition that is equivalent to the one in the book.

**Definition 2.1** Let  $\{\varphi_\sigma\}_{\sigma \in \Sigma^*}$  be an APS (acceptable programming system, i.e., (1)  $\{\varphi_\sigma\}_{\sigma \in \Sigma^*}$  contains exactly the partial recursive functions, and (2) the ‘ $\sigma$ ’ can be manipulated like code). Let  $f$  be universal for  $\{\varphi_\sigma\}_{\sigma \in \Sigma^*}$  (that is,  $f(\sigma, \tau) = \varphi_\sigma(\tau)$ ). The *complexity of  $x$  given  $y$ , relative to  $f$*  is

$$C_f(x|y) = \min\{|\sigma| : f(\sigma, y) = x\}.$$

Note that  $\sigma$  tells how to get from  $y$  to  $x$  since from  $\sigma$  and  $y$  one can produce  $x$ . We also define  $C_f(x) = C_f(x|\lambda)$ . Note that  $C_f(1^n) = \log n + O(1)$  since all you need to describe  $1^n$  is  $n$ .

**Definition 2.2** Let  $c \in \mathbf{N}$  and  $x \in \Sigma^*$ .  $x$  is  *$c$ -incompressible with respect to  $f$*  if  $C_f(x) \geq |x| - c$ . Intuitively we are saying that the shortest description of  $x$  is  $x$  itself. This is a notion of randomness. A simple counting argument shows that most strings are  $c$ -incompressible.

The above definitions seem to depend on the particular  $f$  chosen. The next theorem shows that the definitions are actually robust.

**Theorem 2.3** *Let  $\{\varphi_\sigma\}_{\sigma \in \Sigma^*}$  be an APS and  $f$  be universal for it. Let  $\{\psi_\sigma\}_{\sigma \in \Sigma^*}$  be an APS and  $g$  be universal for it. Then there exists a constant  $c$  such that  $(\forall x)[|C_f(x) - C_g(x)| \leq c]$ . Hence the definition of  $C_f$  is invariant up to an additive constant.*

The approach in the book is to fix one  $f$  and use it to define  $C(x) = C_f(x)$ . Theorem 2.3 shows that results obtained in this way hold for any APS (up to an additive constant). In Chapter 3 they will make a restriction on the APS so that the function  $C$  has some nice properties, but this is the extend to which the APS matters.

In Chapter 2 the authors discuss and prove many properties of  $C(x)$  and incompressible strings. The most interesting nice property is that  $c$ -incompressible strings correspond to another notion of random strings defined by tests. The most interesting negative property is that there is no infinite random string, that is, an infinite string where every prefix is incompressible. Other bad properties of the definition are also noted.

Chapter 3 begins by listing several problems with the definition of  $C(x)$ . The goal of this chapter is to rectify these problems. For this the authors drop their usual verbose style and get to the point:

*Looking at this list of deficiencies with the wisdom of hindsight it is not too difficult to transcend a decade of strenuous investigation and alternative proposals by proceeding immediately to what now seems a convenient version of algorithmic complexity.*

The solution is to make the programs in the APS prefix-free. That is, if  $\sigma$  is a program then  $\sigma\tau$  is not. This restriction of the APS rectifies all the problems that arose. The rest of chapter 3 establishes this and other properties of this definition of complexity. This new notion of complexity is denoted  $K(x)$  instead of  $C(x)$ .

Chapter 4 explores connections between Kolmogorov complexity and probability. Consider the following intuitive problem: getting 20 heads in a row *seems* less likely than getting hthtthtththhhtttthht. If the coins are fair then our intuition is incorrect. However, is there a probability distribution which captures our intuition? We would like a distribution where less complex strings have a lower probability of being observed. The Kolmogorov complexity gives a precise measure of the complexity of a string. Let  $R : \Sigma^* \rightarrow \mathbb{Q}$  be defined by  $R(x) = 2^{-K(x)}$ . This is not a probability distribution, but it does assign to every string some number  $< 1$ , and it does assign larger numbers to more complex strings. (It cannot be made into a distribution by scaling, however its restriction to strings of length  $n$  can.) In this chapter several theorems are proven about  $R(x)$ . The main theorem is that  $R$  is closely related to a universal semi-measure. This is particularly impressive since the definition and construction of universal semi-measures does not use Kolmogorov complexity.

Chapter 5 applies Kolmogorov complexity to inference. Occam's Razor says (informally) that the least complex explanation of a phenomena is the best. Since Kolmogorov complexity gives a precise measure of the complexity of a string, Occam's razor can be made rigorous. This chapter carries this out for Bayesian inference and PAC learning. In most cases the learning procedure is not quite practical; however, some theorems of interest are proven.

Chapter 6 applies Kolmogorov complexity to combinatorics, algorithms analysis, formal language theory, and circuits (the application to circuits is in the second edition but not the first). This is the most interesting chapter for computer scientists who want to see applications. This chapter uses very little Kolmogorov complexity theory: just the existence and abundance of

incompressible strings. We describe some of these applications. There are many more in both the text and the exercises.

1. Let  $r(k)$  be the  $k$ th Ramsey number. Let  $n = k2^{\frac{k}{2}}(\frac{1}{e\sqrt{2}} - o(1))$ . Using the probabilistic method one can show  $r(k) \geq n$ . An alternative proof is to (1) take an incompressible string of length  $\binom{n}{2}$ , (2) use it as the adjacency matrix for a graph  $G$  on  $n$  vertices, and (3) show that  $G$  cannot have a large  $k$ -clique or  $k$ -independent set since otherwise the string that formed  $G$  would have a short description. In this application we obtain an alternative proof of a known theorem. This can be carried out for other probabilistic proofs as well.
2. There are two popular methods for heapsort: William's and Floyd's. It was conjectured many years ago that William's had average running time  $2n \log n - O(n)$  and that Floyd's had average running time  $n \log n - O(n)$ . A proof of this (due to Ian Munroe) is given in the book. The basic idea is to take a permutation  $p$  of  $\{1, \dots, n\}$  such that  $C(p|n) \geq n \log n - 2n$ , and analyze what the algorithm does on  $p$ . Since most permutations have this property, what happens on  $p$  is the average case. An alternative analysis was discovered at roughly the same time as the Kolmogorov proof; however, the Kolmogorov proof is simpler.
3. Let  $L = \{x\#y : x \text{ is a substring of } y\}$ . It is known that a 6-head 2-way DFA can recognize  $L$ . The question arises as to how much this can be scaled down. Using Kolmogorov theory one can show that  $L$  cannot be recognized by a 2-head 1-way DFA (in the text), a 3-head 1-way DFA (as a hard exercise), or a  $k$ -head 1-way DFA with some restrictions. (The unrestricted  $k$ -head case is open.) The only proof known for these results uses Kolmogorov complexity.
4. Håstad's switching lemma says that if you place a random restriction of the variables of a circuit you will, with high probability, be able to switch the bottom two levels. An alternative proof is given where an incompressible string is used rather than a random restriction. This proof seems easier (to me) than the original, but this is a matter of taste.

Chapter 7 is on resource bounded Kolmogorov complexity. Intuitively,  $C^{t,s}(x)$  is the shortest program that prints out  $x$  in time  $t(|x|)$  and space

$s(|x|)$ . This chapter proves theorems of interest about  $C^{t,s}$  and then uses  $C^{t,s}$  to prove theorems in complexity theory. We list some of the theorems.

1. Let  $f, t, s$  be functions,  $n \in \mathbb{N}$ , and  $C[f, t, s] = \{x : |x| = n \wedge C^{t,s}(x) \leq f(n)\}$ . If  $f(n)$  is small (e.g.,  $\sqrt{n}$ ) then this is the set of strings of low resource bounded Kolmogorov complexity. The question arises as to whether there is some tradeoff with the parameters. A very general theorem is proven which says that  $f$  is quite important. One of the corollaries is that, for large  $n$ ,  $C[\sqrt{n} + \log \log n, cn, \infty] - C[\sqrt{n}, \infty, \infty] \neq \emptyset$ .
2.  $A \in \text{BPP}$  (bounded probabilistic polynomial time) if there exists a coin-flipping poly-bounded Turing machine  $M(-; -)$  (the first argument is the input, the second is the sequence of coin flips) such that
 
$$x \in A \Rightarrow \text{Prob}(M(x; \sigma) = 1) \geq 2^{-|x|};$$

$$x \notin A \Rightarrow \text{Prob}(M(x; \sigma) = 0) \geq 2^{-|x|}.$$

If  $x \in \Sigma^*$  and  $\sigma$  is an incompressible string (of the appropriate length) then  $M(x; \sigma) = A(x)$ , otherwise  $\sigma$  is compressible. Hence if one had access to a the function  $C$  then one could easily compute  $A$ . This is not helpful since  $C$  is not recursive. However, using resource bounded Kolmogorov complexity and several other ideas, Sipser (1983 STOC, but also presented in this book) proved that  $\text{BPP} \subseteq \Sigma_2^p \cap \Pi_2^p$ . An easier proof was later found by Lautemann (IPL 1983, Vol 17); however, the ideas behind Sipser's proof may be useful for reasoning about other probabilistic classes.

3. Let  $A \subseteq \Sigma^*$  and  $x \in \Sigma^*$ . Even if  $A$  is hard (e.g., NP-complete) there might be some instances  $x$  that are individually easy. This seems difficult to define because one can always take a program for  $A$  and hard-code  $x$  and  $A(x)$  into it, thus resulting in code for  $A$  that is fast on  $x$ . But such a machine would need size at least  $C^{t,s}(x)$  if you want the machine to run in time  $t$  and space  $s$ , on  $x$ .

Fix time and space bounds  $t$  and  $s$ . The instance complexity of  $x$  (relative to  $A, t$  and  $s$ ) is the size of the least complex program (using  $C^{t,s}$  as your measure of complexity) that is consistent with  $A$  (it may be undefined on some points), defined on  $x$ , and takes time  $t$  and space

s one on all inputs. Several theorems about this notion are proven. (There has been work on instance complexity within recursion theory. This is an exercise on page 501. It is new with the second edition.)

Chapter 8 applies Kolmogorov complexity to Physics. A computation is thought to lose energy; however, if it is reversible then (theoretically) no energy is lost. Hence this chapter spends some pages on reversible computation. In thermodynamics one speaks of the information within a system. Since Kolmogorov complexity gives a rigorous definition of information, connections are made.

### **3 Style**

This book is rather verbose. One can read many pages before getting to the desired information. The book is aware of this and has a special font (also indented) to say “you can skip this part.” Most of these verbose sections are interesting and a pleasant read; however, when looking for a particular item, it can bog you down.

There are many exercises of interest. The book uses the Knuth numbering system to designate the hardness of problems.

### **4 Opinion**

This is a great book. The literature on Kolmogorov complexity is scattered and some of it is in Russian or otherwise hard to access. This book puts it all in one place in a readable, enjoyable style. This is the second edition and the authors have added many new results that have been proven between the first book (in 1993) and this one.