

The truth-table completeness of Kolmogorov complexity

Nils Lauermann

Advisor: Fabian Kunze

Programming Systems Lab
Saarland University

May 20, 2021

History: Kolmogorov complexity

1960

A PRELIMINARY REPORT ON A GENERAL THEORY OF INDUCTIVE INFERENCE

R. J. Solomonoff

Abstract

These preliminary work is presented as a very general new theory of inductive inference. The statement of an infinite sequence of statements is implemented by computing the n -gram probability of random sequences of symbols. The n -gram probability of a sequence is defined by considering a random Turing machine whose output is the sequence in question. An approximation to the n -gram probability is given by the maximal value of the machine that will give the desired result. A more exact formulation is given, and it is made evident that the corresponding probability function will be highly compressible if and only when compressible Turing machines can exist, showing that the machine to be considered has an infinite amount of information in it.

These machines can exist only because the algorithm of the method is a priori problem. Repetition of the method is a priori problem and the corresponding algorithm is a priori problem. These algorithms are of infinite complexity and are presented as a priori problem.

The statement of a random sequence is presented as a priori problem. The statement of a random sequence is presented as a priori problem.

1

“A preliminary report on a general theory of inductive inference.”

Ray J. Solomonoff

1965

International Journal of Computer Mathematics, 1965, Vol. 1, pp. 371-385
© 1965 Gordon and Breach Science Publishers

Three Approaches to the Quantitative Definition of Information*

By A. N. KOLMOGOROV

Mathematics Institute, Vol. 1, No. 3, pp. 3-15, 1965 (Russian version)

There are two common approaches to the quantitative definition of "information": algorithmic and probabilistic. The author treats as well the more formal of these approaches and discusses a new algorithmic approach that uses the theory of measure functions.

1. The Combinatorial Approach

Assume that a variable x is a sequence of values taken in a finite set S containing N elements. We say that the "entropy" of the variable x is

$$H(x) = \log_2 N.$$

By giving x a definite value

$$x = x_0$$

we "consume" this entropy and communicate "information"

$$I = \log_2 N.$$

If the variable $x_1, x_2, \dots, x_n$ are capable of independently taking values in an independently containing $N_1, N_2, \dots, N_n$ members, then

$$H(x_1, x_2, \dots, x_n) = H(x_1) + H(x_2) + \dots + H(x_n). \quad (1)$$

Transmission of a quantity of information I requires

$$I \geq H(x_1, x_2, \dots, x_n).$$

* This note is also known as a problem that the difference $I - H(x)$ is bounded, and $I - H(x)$ is bounded for the value of a probability p .

Received by the publisher January 1965. (Russian version of the paper published in the journal "Mathematics of the USSR", 1965, No. 1, pp. 371-385.)

187

“Three approaches to the quantitative definition of information.”

Andrey N. Kolmogorov

1996

On the Complexity of Random Strings (Extended Abstract)

Martin Kummer

Institut für Informatik, Universität und Fachbereich Informatik, Universität Karlsruhe,
D-76120 Karlsruhe, Germany. kummer@ira.uka.de

Abstract. We show that the set R of Kolmogorov random strings is not a recursive set. This improves the previously known Turing complexity of R and shows how the halting problem can be encoded into the distribution of random strings using the time complexity of non-random strings. As an application we obtain that Post's simple set is truth-table complete in every halting problem. We also show that the truth-table complexity of R cannot be generalized to coin-complexity with respect to arbitrary acceptable countings. It follows we note that R is not frequency computable.

1 Introduction

In Kolmogorov complexity a string x of length n is called random if it is incompressible in the sense that each program which computes x from the empty input has length at least n , i.e., there is no better way to generate x than by table lookup. Let R denote the set of all random strings. Random strings play an important role in many applications of Kolmogorov complexity, most prominently in the "incompressibility method" (see [1, 10] for further details). These bounded versions of randomness are important for theoretical complexity theory and [2, 12] for recent work involving the set R and its time bounded relatives.

In view of the many concrete applications of random strings it is an first glimpse answered surprising that R is a highly non-constructive undecidable set. This is clear consideration reveals that this is inherent in the approach to randomness via incompressibility. The main goal of the present paper is to characterize as precisely as possible the undecidability of R (or equivalently of R in conjunction with the halting problem H).

It is known that R (like set of all non-random strings) is recursively enumerable and Turing complete for the c.e. sets, i.e., the halting problem can be computed using it as an oracle (cf. [11, Theorem 2.10]). But the timing time of the oracle algorithm witnessing this fact is not bounded by any recursive function. The question naturally arises whether there is a recursive algorithm which solves the recursive runtime time. Technically, this question is equivalent to the question whether R is truth-table complete. We give a positive answer and even provide an algorithm of the following type: On input x it computes a finite set R of queries such that $x \in R$ if and only if x is a random string.

Since R is a simple set (i.e., R is infinite and has no infinite recursively enumerable subset), it cannot be bounded truth-table complete, in particular it

“On the complexity of random strings.”

Martin Kummer

Definitions

What is a truth-table reduction?

Many-one reduction

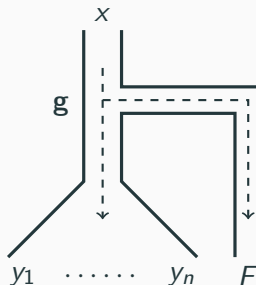
$$f: X \rightarrow Y$$



$$A \leq_m B \Leftrightarrow (x \in A \Leftrightarrow f(x) \in B)$$

Truth-table reduction

$$g: X \rightarrow \mathbb{L}Y \times (\mathbb{L}B \rightarrow \mathbb{B})$$



$$A \leq_{tt} B \Leftrightarrow (x \in A \Leftrightarrow F [\ulcorner y_1 \in B \urcorner, \dots, \ulcorner y_n \in B \urcorner])$$

Basic definitions I

Definition: numbering ψ

$$\psi : \underbrace{\mathbb{N}}_{\text{program}} \times \underbrace{\mathbb{N}}_{\text{input}} \rightarrow \underbrace{\mathbb{N}}_{\text{output}}$$

Definition: Kolmogorov complexity $C_\psi : \mathbb{N} \rightarrow \mathbb{N} \cup \{\infty\}$

$$C_\psi(x) = \min\{\text{length}(p) \mid \psi(p, 0) = x\}$$

Definition: The set of non-random numbers $\overline{R_\psi}$

$$\overline{R_\psi} = \{x \mid C_\psi(x) < \text{length}(x)\}$$

Definition: $M_{n,s}$ - The set of step-limited non-random numbers of length n

$$M_{n,s} = \{\sigma \in \{0,1\}^n \mid \sigma \in \overline{R_\psi} \wedge \psi \text{ terminated in } s \text{ steps}\}$$

$\overline{R_\psi}$ is truth-table complete

Fact: $\overline{R_\psi} \in RE$

For input x try all programs p with $\text{length}(p) < \text{length}(x)$ for increasingly more steps.

Theorem¹: $\forall E \in RE. E \leq_{tt} \overline{R_\psi}$

ψ must be an optimal numbering i.e. can efficiently simulate any other numbering.

¹[Kummer, 1996]

Kummer's proof

Theorem¹: $\forall E \in RE. E \leq_{tt} \overline{R_\psi}$

Definition: E_s

The set of elements enumerated into E before step s

Now, the algorithm to construct the reduction function

¹[Kummer, 1996]

Step 0

$$\begin{array}{c} S_0 \\ \vdots \\ \vdots \\ \vdots \\ S_{i_{max}} \end{array}$$

Step 1

$$\begin{array}{c} S_0 \\ \vdots \\ S_i \\ \vdots \\ S_{i_{\max}} \end{array} \quad \left| \quad \{\dots\} \subseteq \mathbb{N} \right.$$

Step s

S_0	$\{\dots\}$ $\{\dots\}$ $\{\dots\}$
$\vdots$	
S_i	$\{\dots\}$ $\{\dots\}$ $\{\dots\}$ $\{\dots\}$ $\{\dots\}$ $\{\dots\}$ $\{\dots\}$ $\{\dots\}$ $\{\dots\}$
$\vdots$	
$S_{i_{max}}$	$\{\dots\}$ $\{\dots\}$ $\{\dots\}$ $\{\dots\}$

Step s

$$\begin{array}{c|c} S_0 & \{\dots\} \{\dots\} \{\dots\} \\ \vdots & \\ S_i & \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \\ \vdots & \\ S_{i_{\max}} & \{\dots\} \{\dots\} \{\dots\} \{\dots\} \end{array}$$

Choose greatest i ($\leq i_{\max}$) and then smallest n ($\leq s$) s.t.

$$i * c(n) \leq |M_{n,s}|$$

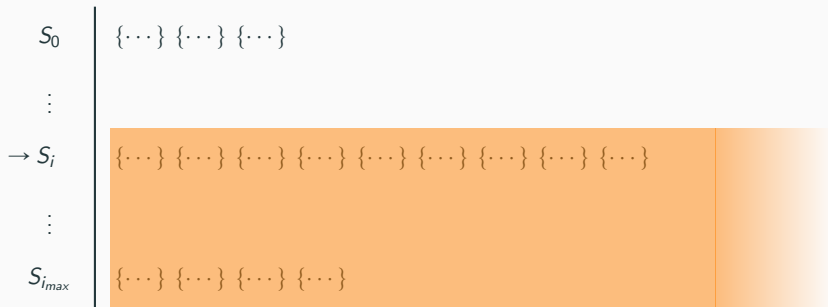
Step s

$$\begin{array}{c|c}
 S_0 & \{\dots\} \{\dots\} \{\dots\} \\
 \vdots & \\
 \rightarrow S_i & \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \\
 \vdots & \\
 S_{i_{\max}} & \{\dots\} \{\dots\} \{\dots\} \{\dots\}
 \end{array}$$

Choose greatest i ($\leq i_{\max}$) and then smallest n ($\leq s$) s.t.

$$i * c(n) \leq |M_{n,s}|$$

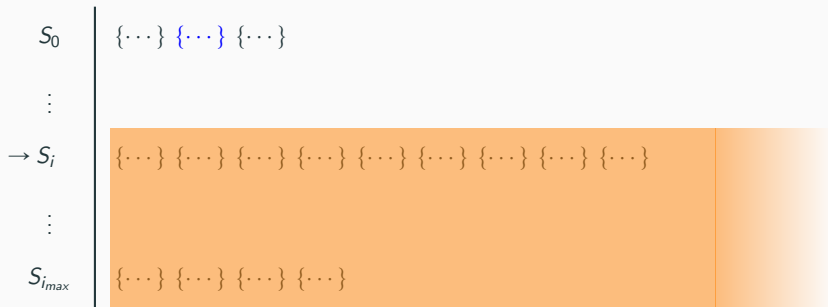
Step s



Choose greatest i ($\leq i_{\max}$) and then smallest n ($\leq s$) s.t.

$$i * c(n) \leq |M_{n,s}| \wedge n \text{ not chosen in } \bullet$$

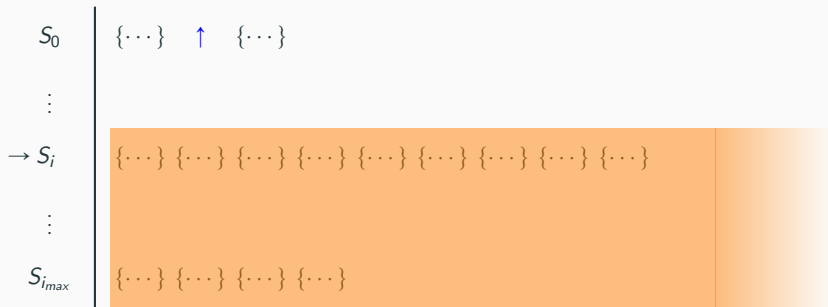
Step s



Choose greatest i ($\leq i_{\max}$) and then smallest n ($\leq s$) s.t.

$$i * c(n) \leq |M_{n,s}| \wedge n \text{ not chosen in } \bullet$$

Step s



Choose greatest i ($\leq i_{\max}$) and then smallest n ($\leq s$) s.t.

$$i * c(n) \leq |M_{n,s}| \wedge n \text{ not chosen in } \bullet$$

Step s

S_0	$\{\dots\} \quad \uparrow \quad \{\dots\}$
$\vdots$	
$\rightarrow S_i$	$\{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} S_{i,k}$
$\vdots$	
$S_{i_{max}}$	$\{\dots\} \{\dots\} \{\dots\} \{\dots\}$

Choose greatest i ($\leq i_{max}$) and then smallest n ($\leq s$) s.t.

$$i * c(n) \leq |M_{n,s}| \wedge n \text{ not chosen in } \bullet$$

Step s

$$\begin{array}{c|c}
 S_0 & \{\dots\} \quad \uparrow \quad \{\dots\} \\
 \vdots & \\
 \rightarrow S_i & \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} S_{i,k} \\
 \vdots & \\
 S_{i_{\max}} & \{\dots\} \{\dots\} \{\dots\} \{\dots\}
 \end{array}$$

Choose greatest i ($\leq i_{\max}$) and then smallest n ($\leq s$) s.t.

$$i * c(n) \leq |M_{n,s}| \wedge n \text{ not chosen in } \bullet$$

$$S_{i,k} := c(n) \text{ smallest elements of } \{0,1\}^n - M_{n,s}$$

Step s

$$\begin{array}{c|c}
 S_0 & \{\dots\} \quad \uparrow \quad \{\dots\} \\
 \vdots & \\
 S_i & \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} S_{i,k} \\
 \vdots & \\
 S_{i_{\max}} & \{\dots\} \{\dots\} \{\dots\} \{\dots\}
 \end{array}$$

Choose greatest i ($\leq i_{\max}$) and then smallest n ($\leq s$) s.t.

$$i * c(n) \leq |M_{n,s}| \wedge n \text{ not chosen in } \bullet$$

Step s

S_0	$\{\dots\} \quad \uparrow \quad \{\dots\}$
$\vdots$	
S_i	$\{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} S_{i,k}$
$\vdots$	
$S_{i_{max}}$	$\{\dots\} \{\dots\} \{\dots\} \{\dots\}$

Choose greatest i ($\leq i_{max}$) and then smallest n ($\leq s$) s.t.

$$i * c(n) \leq |M_{n,s}| \wedge n \text{ not chosen in } \bullet \wedge n \text{ unlocked}$$

For any j & x with $x \in E_s \wedge S_{j,x} \downarrow$ do: lock the n of $S_{j,x}$

Step s

S_0	$\{\dots\} \quad \uparrow \quad \{\dots\}$
$\vdots$	
S_i	$\{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} S_{i,k}$
$\vdots$	
$S_{i_{max}}$	$\{\dots\} \{\dots\} \{\dots\} \{\dots\}$

Choose greatest i ($\leq i_{max}$) and then smallest n ($\leq s$) s.t.

$$i * c(n) \leq |M_{n,s}| \wedge n \text{ not chosen in } \bullet \wedge n \text{ unlocked}$$

For any j & x with $x \in E_s \wedge S_{j,x} \downarrow$ do: lock the n of $S_{j,x}$ and **force** $S_{j,x} \subseteq \overline{R_\psi}$

Step ∞

S_0	$\{\dots\} \{\dots\} \{\dots\}$
$\vdots$	
S_i	$\{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\}$
$\vdots$	
$S_{i_{max}}$	$\{\dots\} \{\dots\} \{\dots\} \{\dots\}$

Choose greatest i ($\leq i_{max}$) and then smallest n ($\leq s$) s.t.

$$i * c(n) \leq |M_{n,s}| \wedge n \text{ not chosen in } \bullet \wedge n \text{ unlocked}$$

Step ∞

S_0	$\{\dots\} \{\dots\} \{\dots\}$
$\vdots$	
S_i	$\{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \dots$
$\vdots$	
$S_{i_{max}}$	$\{\dots\} \{\dots\} \{\dots\} \{\dots\}$

Choose greatest i ($\leq i_{max}$) and then smallest n ($\leq s$) s.t.

$$i * c(n) \leq |M_{n,s}| \wedge n \text{ not chosen in } \bullet \wedge n \text{ unlocked}$$

Step ∞

$$\begin{array}{c|l}
 S_0 & \{\dots\} \{\dots\} \{\dots\} \\
 \vdots & \\
 S_i & \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \dots \\
 \vdots & \\
 S_{i_{\max}} & \{\dots\} \{\dots\} \{\dots\} \{\dots\}
 \end{array}
 \left. \vphantom{\begin{array}{c} S_0 \\ \vdots \\ S_i \\ \vdots \\ S_{i_{\max}} \end{array}} \right\} \text{finite}$$

Choose greatest i ($\leq i_{\max}$) and then smallest n ($\leq s$) s.t.

$$i * c(n) \leq |M_{n,s}| \wedge n \text{ not chosen in } \bullet \wedge n \text{ unlocked}$$

Step ∞

$$\begin{array}{c|l}
 S_0 & \{\dots\} \{\dots\} \{\dots\} \\
 \vdots & \\
 S_{i_0} & \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \dots \\
 \vdots & \\
 S_{i_{\max}} & \{\dots\} \{\dots\} \{\dots\} \{\dots\}
 \end{array}
 \left. \vphantom{\begin{array}{c} S_0 \\ \vdots \\ S_{i_0} \\ \vdots \\ S_{i_{\max}} \end{array}} \right\} \text{finite}$$

Choose greatest i ($\leq i_{\max}$) and then smallest n ($\leq s$) s.t.

$$i * c(n) \leq |M_{n,s}| \wedge n \text{ not chosen in } \bullet \wedge n \text{ unlocked}$$

Step ∞

$$\begin{array}{c|l}
 S_0 & \{\dots\} \{\dots\} \{\dots\} \\
 \vdots & \\
 S_{i_0} & \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \{\dots\} \dots \\
 \vdots & \\
 S_{i_{\max}} & \{\dots\} \{\dots\} \{\dots\} \{\dots\} \left. \vphantom{\begin{array}{c} \vdots \\ S_{i_0} \end{array}} \right\} \text{finite}
 \end{array}$$

Choose greatest i ($\leq i_{\max}$) and then smallest n ($\leq s$) s.t.

$$i * c(n) \leq |M_{n,s}| \wedge n \text{ not chosen in } \bullet \wedge n \text{ unlocked}$$

We now show $\exists x'. \forall x > x'. x \in E \Leftrightarrow S_{i_0,x} \subseteq \overline{R_\psi}$

Why does that work?

$$\exists x'. \forall x > x'. x \in E \Leftrightarrow S_{i_0, x} \subseteq \overline{R_\psi}$$

What is x' ?

The least index s.t. after the definition of $S_{i_0, x'}$ no S_j ($j > i_0$) gets extended

$\Rightarrow$: We know

- $\exists s. x \in E_s$
- $S_{i_0, x}$ is almost always defined

Reminder: The end of each step

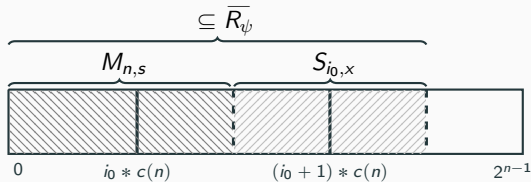
$\forall jx. x \in E_s \wedge S_{j, x} \downarrow$ do :

- force $S_{j, x} \subseteq \overline{R_\psi}$
- lock n

$$\Rightarrow S_{i_0, x} \subseteq \overline{R_\psi}$$

Why does that work?

$\Leftarrow$: We know $S_{i_0,x} \subseteq \overline{R_\psi}$



- $\exists s'. (i_0 + 1) * c(n) \leq |M_{n,s'}|$
- i_0 not greatest
unless n gets locked beforehand

$\Rightarrow x \in E$

Reminder:

- $S_{i_0,x} := c(n)$ smallest elements of $\{0, 1\}^n - M_{n,s}$
- At the start of each step: Choose greatest i s.t.
 $i * c(n) \leq |M_{n,s}| \wedge n$ unlocked $\wedge \dots$
- At the end of each step: $\forall jx. x \in E_s \wedge S_{j,x} \downarrow$ do :
 - force $S_{j,x} \subseteq \overline{R_\psi}$
 - lock n

The reduction: $E \leq_{tt} \overline{R_\psi}$

$$f : X \rightarrow \mathbb{LN} \times (\mathbb{LB} \rightarrow \mathbb{B})$$

$$f(x) = \begin{cases} (\emptyset, \lambda a.\text{true}) & x < x' \wedge x \in E \\ (\emptyset, \lambda a.\text{false}) & x < x' \wedge x \notin E \\ (S_{i_0, x}, \wedge) & x \geq x' \end{cases}$$

Conclusion

- The intricacies of Kummer's proof (not discussed here) make it hard to comprehend
- The proof idea is used in other related proofs
- The non-constructiveness will be a major challenge of the formalization

The road ahead:

- deciding on suitable definitions in Coq
- proving the results in the most straightforward way
- (beautifying as far as possible)

Thank you!

References

References

-  Kolmogorov, A. N. (1965).
Three approaches to the quantitative definition of information’.
Problems of information transmission, 1(1):1–7.
-  Kummer, M. (1996).
On the complexity of random strings.
In Puech, C. and Reischuk, R., editors, *STACS 96*, pages 25–36. Springer Berlin Heidelberg.
-  Solomonoff, R. J. (1960).
A preliminary report on a general theory of inductive inference.
Citeseer.