# Kolmogorov Complexity

[Kolmogorov Complexity 1](#_Toc2115506003)

[Definition, Terms, and Conceptual Affinities 1](#_Toc1865046531)

[Occam's Razor 2](#_Toc514162179)

[Shannon Entropy 2](#_Toc448240656)

[Difference from Kolmogorov: 3](#_Toc1736982287)

[The difference between both Shannon entropy and Kolmogorov complexity with a numerical example : 3](#_Toc1922133676)

[Shannon Entropy 3](#_Toc182544089)

[Kolmogorov Complexity 4](#_Toc1429172327)

[Kolmogorov Complexity and DNA Analysis 4](#_Toc1290222753)

[Application Method (Simplified) 5](#_Toc1078741522)

[Mathematical definition of Kolmogorov complexity 6](#_Toc239215216)

[Uncomputability of Kolmogorov Complexity 6](#_Toc687263091)

[Halting Problem 7](#_Toc5693013)

[Collatz sequence 7](#_Toc105940118)

[Randomness According to Kolmogorov — Mathematical Definition 8](#_Toc1214318928)

[Is the Number π Random? 8](#_Toc1925252716)

[REFERENCES 9](#_Toc356901692)

## Definition, Terms, and Conceptual Affinities

*“The complexity of an object is the length of the shortest binary program that outputs this object.”*

**Definition**: Kolmogorov complexity is defined as the length of the shortest statement (program) required to produce a data or string.

This approach aims to measure the inherent complexity of a piece of information based on its content rather than its probabilities.

**Origin:**

Developed by **Andrey N. Kolmogorov** in the mid-1960s.

Shannon entropy, previously used in information theory, measured the average uncertainty in a distribution, but it could not describe the complexity of individual (unique) strings.

Kolmogorov developed this concept to describe "how much information" any single string contains.

**Kolmogorov complexity allows:**

* to analyze the compressibility of data,
* to measure the degree of randomness, and to mathematically express the principle of "saying a lot with a few words" (Occam's razor).

## Occam's Razor

*“If two hypotheses explain the data equally well, choose the simpler one.”*

**Definition:**

Occam's razor is a principle of simplicity used in science and philosophy.

**It states:**

“If there is more than one hypothesis explaining the same phenomenon, the simplest one should be preferred.”

That is, unnecessary complexity and extraneous assumptions should be avoided.

**Relation to Kolmogorov:**

Kolmogorov complexity quantifies this principle (mathematically):

Strings with shorter explanations are considered “simpler.”

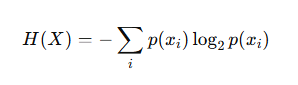
## Shannon Entropy

*“The entropy of a discrete source is a measure of the average information content per symbol.”*

**Definition:**

Shannon entropy measures the average ambiguity of symbols from an information source.

By formula ,



Here, 𝑝(𝑥𝑖)

is the probability of each symbol.

The higher the entropy, the more unpredictable (uncertain) the symbols are and the higher their information content.

It requires a probability distribution as input (e.g., the probability of each letter appearing).

According to this distribution, it returns the average amount of information over the long term.

### Difference from Kolmogorov:

Shannon entropy → average amount of information, based on probability distribution

Kolmogorov complexity → shortest description length, based on a single string itself

### The difference between both Shannon entropy and Kolmogorov complexity with a numerical example :

Two Strings

* **String A: 00000000**
* **String B: 01011010**

Each string is 8 bits long.

### Shannon Entropy

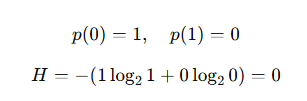
Shannon entropy looks at a probability distribution.

Let's count the proportions of symbols in strings:

String A : **00000000**

Number 0: 8

Number 1: 0



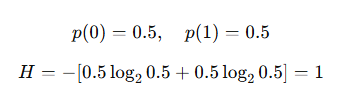
Entropy = 0 bits/symbol → **No uncertainty**

String B : **01011010**

Number 0: 4

Number 1: 4

𝑝(0)=0.5,𝑝(1)=0.5



Entropy = 1 bit/symbol → Maximum uncertainty

### Kolmogorov Complexity

Kolmogorov complexity looks at the shortest description length of a single string:

**String A: 00000000**

→ “Repeat 0 8 times” → very short description → low complexity

**String B: 01011010**

→ No clear pattern → only way to write every bit → high complexity

**This allows us to:**

Measure whether a single string contains a pattern.

Analyze the compressibility of a string.

Answer the question, "Is this data truly random or does it follow a rule?"

## Kolmogorov Complexity and DNA Analysis

**The basic idea:**

DNA sequences are very long strings of characters (consisting of the letters A, T, G, and C).

The genome of each species or individual exhibits different regularities and repetition rates.

Kolmogorov complexity provides the answer to this question:

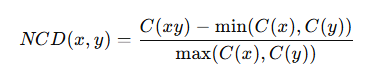
"How regular (simple) or random (complex) does this DNA sequence appear?"

### Application Method (Simplified)

Because true Kolmogorov complexity is incomputable, the following approach is used:

* A DNA sequence (e.g., a 3-million-base genome) is taken.
* This string is compressed using compression algorithms (gzip, bzip2, LZ77, DNA-specific compressors).
* The size of the compressed file is considered an approximation of the Kolmogorov complexity.
* If it compresses more → regular → low complexity
* If it doesn't compress → random → high complexity
* This method is called Normalized Compression Distance (NCD) and is based on Kolmogorov complexity theory.

To measure this difference, the Normalized Compression Distance (NCD) formula is usually used.



𝐶(𝑥): Compressed length of array x

𝐶(𝑦): Compressed length of array y

𝐶(𝑥𝑦): Compressed length of the combination of x and y

Low NCD value: Two strings are very similar → there are many common repetitions.

High NCD value: Two strings are very different → there are few common repetitions.

## Mathematical definition of Kolmogorov complexity

Setup (notation):

Let {0,1} be the set of all finite binary strings.

Let 𝑈 be a fixed universal Turing machine (UTM).

∣𝑝∣: program length of 𝑝 in bits.

x ∈ {0,1} — a finite binary string (output),

𝑝 — a binary program (input to 𝑈),

∣𝑝 ∣ — length of program 𝑝 in bits,

𝑈(𝑝)=𝑥 — running 𝑝 on 𝑈 halts and outputs 𝑥.

Plain Kolmogorov complexity

The plain Kolmogorov complexity of a string 𝑥 ∈ {0,1} is:





That is:

The Kolmogorov complexity of 𝑥 is the length of the shortest binary program 𝑝 that outputs 𝑥 when executed 𝑈.

## Uncomputability of Kolmogorov Complexity

Although Kolmogorov complexity is **mathematically well-defined**, it is **not computable**.

*"𝐶(𝑥) is not a computable function. Otherwise, the halting problem could be solved, which is impossible."*

To calculate 𝐶(𝑥), we must try all possible programs that produce 𝑥 and choose the shortest one.

However, some programs never stop, and determining which program will stop requires solving the stopping problem.

Since the stopping problem is unsolvable,

𝐶(𝑥) is also uncomputable.

### Halting Problem

**Definition:**

The halting problem asks the following question for any program 𝑃 and input 𝑤:

“Does 𝑃(𝑤) stop when it is executed, or does it run forever ?”

def paradox(program):

if is\_stop(program, program):

while True: # Run forever

pass

else:

return

Paradox Analysis

Consider the call to paradox:

Scenario 1: If paradox STOPS

is\_stop (paradox, paradox) → returns True

Then while True executes

The program DOES NOT STOP CONTRADICTION!

Scenario 2: If paradox STOPS

is\_stop(paradox, paradox) → returns False

Then return executes

The program DOES STOP CONTRADICTION!

In both cases, there is a contradiction! Therefore, the function is\_stop cannot exist.

### Collatz sequence

def program3(n):

while n != 1:

if n % 2 == 0:

n = n // 2

else:

n = 3 \* n + 1

return

This is a Collatz sequence problem. For n=7:

7 → 22 → 11 → 34 → 17 → 52 → 26 → 13 → 40 → 20 → 10 → 5 → 16 → 8 → 4 → 2 → 1 (STOPS)

But it remains to be proven that it will stop for all values of n!

## Randomness According to Kolmogorov — Mathematical Definition

Definition (**Martin-Löf–Kolmogorov approximation**):

A binary string 𝑥 is considered random if:

The length of the shortest program that generates it is almost equal to the length of the string itself.

Formally:



Here:

𝐾(𝑥) : Kolmogorov (prefix) complexity of 𝑥

∣𝑥∣ : length of the string (in bits)

𝑐 : a small constant

If a string has a compressible short description → low complexity → non-random

If it cannot be compressed at all → high complexity → random

This is Kolmogorov's definition of randomness.

## Is the Number π Random?

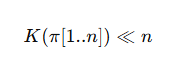
No, it is not random in the Kolmogorov sense.

Why?

The decimal or binary expansion of π appears very complex and irregular, but:

There is a known short algorithm for calculating π (e.g., the Bailey–Borwein–Plouffe formula).

So, the length of a program that generates the first 𝑛 digits of π is much shorter than that of 𝑛.



Because π is apparently complex but compressible, according to Kolmogorov, it is not random .

Such numbers are called deterministic complex sequences (deterministic but complex).

**Example of True Randomness**

String formed by flipping a coin 𝑛 times .Most of these strings have a Kolmogorov complexity of ≈ 𝑛 .So, their shortest explanation is themselves

→ They are Kolmogorov random.

## REFERENCES

* Kolmogorov, A. N. (1965). *Three approaches to the quantitative definition of information*. Problems of Information Transmission.
* Li, M., & Vitányi, P. (2008). *An Introduction to Kolmogorov Complexity and Its Applications*
* Martin-Löf, P. (1966). *The definition of random sequences*. Information and Control
* Calude, C. (2002). *Information and Randomness: An Algorithmic Perspective*
* Turing, A. M. (1936). *On computable numbers, with an application to the Entscheidungsproblem*.