Kolmogorov Complexity in perspective. Part I: Information Theory and Randomnes

Marie Ferbus-Zanda, Serge Grigorieff

Kolmogorov Complexity in perspective

Part I: Information Theory and Randomness[*]

Marie Ferbus-Zanda
LIAFA, CNRS & Université Paris 7
case 7014
75205 Paris Cedex 13 France
Marie.Ferbus@liafa.jussieu.fr

Serge Grigorieff
LIAFA, CNRS & Université Paris 7
case 7014
75205 Paris Cedex 13 France
seg@liafa.jussieu.fr

Abstract

We survey diverse approaches to the notion of information: from Shannon entropy to Kolmogorov complexity. Two of the main applications of Kolmogorov complexity are presented: randomness and classification. The survey is divided in two parts in the same volume.

Part I is dedicated to information theory and the mathematical formalization of randomness based on Kolmogorov complexity. This last application goes back to the 60's and 70's with the work of Martin-Löf, Schnorr, Chaitin, Levin, and has gained new impetus in the last years.

Keywords: Logic, Computer Science, Algoritmmic Information Theory, Shannon Information Theory, Kolmogorov Complexity, Randomness.

Contents

[*] Published in Synthese, 2008.

Note. Following Robert Soare's recommendations ([49], 1996), which have now gained large agreement, we write *computable* and *computably enumerable* in place of the old fashioned *recursive* and *recursively enumerable*.

Notation. By $\log x$ (resp. $\log_s x$) we mean the logarithm of x in base 2 (resp. base s where $s \geq 2$). The "floor" and "ceil" of a real number x are denoted by $\lfloor x \rfloor$ and $\lceil x \rceil$: they are respectively the largest integer $\leq x$ and the smallest integer $\geq x$. Recall that, for $s \geq 2$, the length of the base s representation of an integer k is $\ell \geq 1$ if and only if $s^{\ell-1} \leq k < s^{\ell}$. Thus, the length of the base s representation of an integer k is $1 + \lfloor \log_s k \rfloor = 1 + \lfloor \frac{\log k}{\log s} \rfloor$.
The number of elements of a finite family $\mathcal{F}$ is denoted by $\sharp\mathcal{F}$.
The length of a word u is denoted by $|u|$.

1 Three approaches to a quantitative definition of information

A title borrowed from Kolmogorov' seminal paper ([25], 1965).

1.1 Which information?

1.1.1 About anything...

About anything can be seen as conveying information. As usual in mathematical modelization, we retain only a few features of some real entity or process, and associate to them some finite or infinite mathematical objects. For instance,

- - an integer or a rational number or a word in some alphabet,
  - a finite sequence or a finite set of such objects,
  - a finite graph,...

- - a real,
  - a finite or infinite sequence of reals or a set of reals,
  - a function over words or numbers,...

This is very much as with probability spaces. For instance, to modelize the distributions of 6 balls into 3 cells, (cf. Feller, [15], §I.2, II.5) we forget everything about the nature of balls and cells and of the distribution process, retaining only two questions: "how many balls in each cell?" and "are the balls and cells distinguishable or not?". Accordingly, the modelization considers
- either the $729 = 3^6$ maps from the set of balls into the set of cells in case the balls are distinguishable and so are the cells (this is what is done in Maxwell-Boltzman statistics),
- or the $28 = \begin{pmatrix} 6+(3-1) \\ 6 \end{pmatrix}$ triples of non negative integers with sum[1] 6 in

[1] This value is easily obtained by identifying such a triple with a binary word with six letters 0 for the six balls and two letters 1 to mark the partition in the three cells.

case the cells are distinguishable but not the balls (this is what is done in Bose-Einstein statistics)
- or the 7 sets of at most 3 integers with sum 6 in case the balls are undistinguishable and so are the cells.

1.1.2 Especially words

In information theory, special emphasis is made on information conveyed by words on finite alphabets. I.e., on *sequential information* as opposed to the obviously massively parallel and interactive distribution of information in real entities and processes. A drastic reduction which allows for mathematical developments (but also illustrates the Italian saying "traduttore, traditore!").

As is largely popularized by computer science, any finite alphabet with more than two letters can be reduced to one with exactly two letters. For instance, as exemplified by the ASCII code (American Standard Code for Information Interchange), any symbol used in written English – namely the lowercase and uppercase letters, the decimal digits, the diverse punctuation marks, the space, apostrophe, quote, left and right parentheses – together with some simple typographical commands – such as tabulation, line feed, carriage return or "end of file" – can be coded by binary words of length 7 (corresponding to the 128 ASCII codes). This leads to a simple way to code any English text by a binary word (which is 7 times longer)[2].

Though quite rough, the length of a word is the basic measure of its information content. Now, a fairness issue faces us: richer the alphabet, shorter the word. Considering groups of k successive letters as new letters of a super-alphabet, one trivially divides the length by k. For instance, a length n binary word becomes a length $\lceil \frac{n}{256} \rceil$ word with the usual packing of bits by groups of 8 (called bytes) which is done in computers.
This is why all considerations about the length of words will always be developed relative to binary alphabets. A choice to be considered as a *normalization of length.*

Finally, we come to the basic idea to measure the information content of a mathematical object x :

$$\boxed{\textit{information content of } x \quad = \quad \begin{array}{l} \textit{length of a shortest binary word} \\ \textit{which “encodes” } x \end{array}}$$

What do we mean precisely by "encodes" is the crucial question. Following the trichotomy pointed by Kolmogorov in [25], 1965, we survey three approaches.

[2] For other European languages which have a lot of diacritic marks, one has to consider the 256 codes of Extended ASCII which have length 8. And for non European languages, one has to turn to the 65 536 codes of Unicode which have length 16.

1.2 Combinatorial approach: entropy

1.2.1 Constant-length codes

Let us consider the family A^n of length n words in an alphabet A with s letters $a_1, ..., a_s$. Coding the a_i's by binary words w_i's all of length $\lceil \log s \rceil$, to any word u in A^n we can associate the binary word ξ obtained by substituting the w_i's to the occurrences of the a_i's in u. Clearly, ξ has length $n\lceil \log s \rceil$. Also, the map $u \mapsto \xi$ from the set A^* of words in alphabet A to the set $\{0,1\}^*$ of binary words is very simple. Mathematically, considering on A^* and $\{0,1\}^*$ the algebraic structure of monoid given by the concatenation product of words, this map $u \mapsto \xi$ is a morphism since the image of a concatenation uv is the concatenation of the images of u and v.

1.2.2 Variable-length prefix codes

Instead of coding the s letters of A by binary words of length $\lceil \log s \rceil$, one can code the a_i's by binary words w_i's having different lengthes so as to associate short codes to most frequent letters and long codes to rare ones. This is the basic idea of compression. Using such codes, the substitution of the w_i's to the occurrences of the a_i's in a word u gives a binary word ξ. And the map $u \mapsto \xi$ is again very simple. It is still a morphism from the monoid of words on alphabet A to the monoid of binary words and can also be computed by a finite automaton.

Now, we face a problem: can we recover u from ξ ? i.e., is the map $u \mapsto \xi$ injective? In general the answer is no. However, a simple sufficient condition to ensure decoding is that the family $w_1, ..., w_s$ be a so-called *prefix-free code* (or *prefix code*). Which means that if $i \neq j$ then w_i is not a prefix of w_j.

> This condition insures that there is a unique w_{i_1} which is a prefix of ξ. Then, considering the associated suffix ξ_1 of v (i.e., $v = w_{i_1}\xi_1$) there is a unique w_{i_2} which is a prefix of ξ_1, i.e., u is of the form $u = w_{i_1}w_{i_2}\xi_2$. And so on.

Suppose the numbers of occurrences in u of the letters $a_1, ..., a_s$ are $m_1, ..., m_s$, so that the length of u is $n = m_1 + ... + m_s$. Using a prefix-free code $w_1, ..., w_s$, the binary word ξ associated to u has length $m_1|w_1| + ... + m_s|w_s|$. A natural question is, given $m_1, ..., m_s$, *how to choose the prefix-free code* $w_1, ..., w_s$ *so as to minimize the length of* ξ *?*
Huffman ([21], 1952) found a very efficient algorithm (which has linear time complexity if the frequencies are already ordered). This algorithm (suitably modified to keep its top efficiency for words containing long runs of the same data) is nowadays used in nearly every application that involves the compression and transmission of data: fax machines, modems, networks,...

1.2.3 Entropy of a distribution of frequencies

The intuition of the notion of entropy in information theory is as follows. Given natural integers $m_1, ..., m_s$, consider the family $\mathcal{F}_{m_1,...,m_s}$ of length $n = m_1 + ... + m_s$ words of the alphabet A in which there are exactly $m_1, ..., m_s$ occurrences of letters $a_1, ..., a_s$. How many binary digits are there in the binary representation of the number of words in $\mathcal{F}_{m_1,...,m_s}$? It happens (cf. Proposition 1.2) that this number is essentially linear in n, the coefficient of n depending solely on the frequencies $\frac{m_1}{n}, ..., \frac{m_s}{n}$. It is this coefficient which is called the entropy H of the distribution of the frequencies $\frac{m_1}{n}, ..., \frac{m_s}{n}$.

Definition 1.1 (Shannon, [48], 1948)**.** *Let $f_1, ..., f_s$ be a distribution of frequencies, i.e., a sequence of reals in $[0,1]$ such that $f_1 + ... + f_s = 1$. The entropy of $f_1, ..., f_s$ is the real*

$$H = -(f_1 \log(f_1) + ... + f_s \log(f_s))$$

Proposition 1.2 (Shannon, [48], 1948)**.** *Let $m_1, ..., m_s$ be natural integers and $n = m_1 + ... + m_s$. Then, letting H be the entropy of the distribution of frequencies $\frac{m_1}{n}, ..., \frac{m_s}{n}$, the number $\sharp\mathcal{F}_{m_1,...,m_s}$ of words in $\mathcal{F}_{m_1,...,m_s}$ satisfies*

$$\log(\sharp\mathcal{F}_{m_1,...,m_s}) = nH + O(\log n)$$

where the bound in $O(\log n)$ depends solely on s and not on $m_1, ..., m_s$.

> *Proof.* The set $\mathcal{F}_{m_1,...,m_s}$ contains $\frac{n!}{m_1! \times ... \times m_s!}$ words. Using Stirling's approximation of the factorial function (cf. [15]), namely $x! = \sqrt{2\pi}\ x^{x+\frac{1}{2}}\ e^{-x+\frac{\theta}{12}}$ where $0 < \theta < 1$, and equality $n = m_1 + ... + m_S$, we get
>
> $$\begin{aligned} \log(\frac{n!}{m_1! \times ... \times m_s!}) &= (\sum_i m_i) \log(n) - (\sum_i m_i \log m_i) \\ &\quad + \frac{1}{2} \log(\frac{n}{m_1 \times ... \times m_s}) - (s-1) \log \sqrt{2\pi} + \alpha \end{aligned}$$
>
> where $|\alpha| \leq \frac{s}{12} \log e$. The difference of the first two terms is equal to $n[\sum_i \frac{m_i}{n} \log(\frac{m_i}{n})] = nH$ and the remaining sum is $O(\log n)$ since $n^{1-s} \leq \frac{n}{m_1 \times ... \times m_s} \leq n$. □

H has a striking significance in terms of information content and compression. Any word u in $\mathcal{F}_{m_1,...,m_s}$ is uniquely characterized by its rank in this family (say relatively to the lexicographic ordering on words in alphabet A). In particular, the binary representation of this rank "encodes" u. Since this rank is $< \sharp\mathcal{F}_{m_1,...,m_s}$, its binary representation has length $\leq nH$ up to an $O(\log n)$ term. Thus, nH can be seen as an upper bound of the information content of u. Otherwise said, the n letters of u are encoded by nH binary digits. In terms of compression (nowadays so popular with the zip-like softwares), *u can be compressed to nH bits*, i.e., *the mean information content (which can be seen*

as the compression size in bits) of a letter of u is H.
Let us look at two extreme cases.
• If all frequencies f_i are equal to $\frac{1}{s}$ then the entropy is $\log(s)$, so that the mean information content of a letter of u is $\log(s)$, i.e., there is no better (prefix-free) coding than that described in §1.2.1.
• In case some of the frequencies is 1 (hence all other ones being 0), the information content of u is reduced to its length n, which, written in binary, requires $\log(n)$ bits. As for the entropy, it is 0 (with the usual convention $0 \log 0 = 0$, justified by the fact that $\lim_{x \to 0} x \log x = 0$). The discrepancy between $nH = 0$ and the true information content $\log n$ comes from the $O(\log n)$ term in Proposition1.2.

1.2.4 Shannon's source coding theorem for symbol codes

The significance of the entropy explained above has been given a remarkable and precise form by Claude Elwood Shannon (1916-2001) in his celebrated paper [48], 1948. It's about the length of the binary word ξ associated to u via a prefix-free code. Shannon proved
- a lower bound of $|\xi|$ valid whatever be the prefix-free code $w_1, ..., w_s$,
- an upper bound, quite close to the lower bound, valid for particular prefix-free codes $w_1, ..., w_s$ (those making ξ shortest possible, for instance those given by Huffman's algorithm).

Theorem 1.3 (Shannon, [48], 1948)**.** *Suppose the numbers of occurrences in u of the letters $a_1, ..., a_s$ are $m_1, ..., m_s$. Let $n = m_1 + ... + m_s$.*

1. For every prefix-free sequence of binary words $w_1, ..., w_s$ (which are to code the letters $a_1, ..., a_s$), the binary word ξ obtained by substituting w_i to each occurrence of a_i in u satisfies

$$nH \leq |\xi|$$

where $H = -(\frac{m_1}{n} \log(\frac{m_1}{n}) + ... + \frac{m_s}{n} \log(\frac{m_s}{n}))$ is the entropy of the considered distribution of frequencies $\frac{m_1}{n}, ..., \frac{m_s}{n}$.

2. There exists a prefix-free sequence of binary words $w_1, ..., w_s$ such that

$$nH \leq |\xi| < n(H + 1)$$

Proof. First, we recall two classical results.

Kraft's inequality. Let $\ell_1, ..., \ell_s$ be a finite sequence of integers. Inequality $2^{-\ell_1} + ... + 2^{-\ell_s} \leq 1$ holds if and only if there exists a prefix-free sequence of binary words $w_1, ..., w_s$ such that $\ell_1 = |w_1|, ..., \ell_s = |w_s|$.

Gibbs' inequality. Let $p_1, ..., p_s$ and $q_1, ..., q_s$ be two probability distributions, i.e., the p_i's (resp. q_i's) are in $[0, 1]$ and have sum 1. Then $-\sum p_i \log(p_i) \leq -\sum p_i \log(q_i)$ with equality if and only if $p_i = q_i$ for all i.

Proof of Point 1 of Theorem 1.3. Set $p_i = \frac{m_i}{n}$ and $q_i = \frac{2^{-|w_i|}}{S}$ where $S = \sum_i 2^{-|w_i|}$. Then

$$|\xi| = \sum_i m_i |w_i| = n[\sum_i \frac{m_i}{n}(-\log(q_i) - \log S)]$$
$$\geq n[-(\sum_i \frac{m_i}{n}\log(\frac{m_i}{n}) - \log S] = n[H - \log S] \geq nH$$

The first inequality is an instance of Gibbs' inequality. For the last one, observe that $S \leq 1$.

Proof of Point 2 of Theorem 1.3. Set $\ell_i = \lceil -\log(\frac{m_i}{n})\rceil$. Observe that $2^{-\ell_i} \leq \frac{m_i}{n}$. Thus, $2^{-\ell_1} + ... + 2^{-\ell_s} \leq 1$. Applying Kraft inequality, we see that there exists a prefix-free family of words $w_1, ..., w_s$ with lengthes $\ell_1, ..., \ell_s$.

We consider the binary word ξ obtained via this prefix-free code, i.e., ξ is obtained by substituting w_i to each occurrence of a_i in u. Observe that $-\log(\frac{m_i}{n}) \leq \ell_i < -\log(\frac{m_i}{n}) + 1$. Summing, we get $nH \leq |\xi| < n(H + 1)$. □

In particular cases, the lower bound nH can be achieved.

Theorem 1.4. *In case the frequencies $\frac{m_i}{n}$'s are all negative powers of two (i.e., $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}$,...) then the optimal ξ (given by Huffman's algorithm) satisfies $\xi = nH$.*

1.2.5 Closer to the entropy

In §1.2.3 and 1.2.4, we supposed the frequencies to be known and did not consider the information content of these frequencies. We now deal with that question.

Let us go back to the encoding mentioned at the start of §1.2.3. A word u in the family $\mathcal{F}_{m_1,...,m_s}$ (of length n words with exactly $m_1, ..., m_s$ occurrences of $a_1, ..., a_s$) can be recovered from the following data:

- the values of $m_1, ..., m_s$,
- the rank of u in $\mathcal{F}_{m_1,...,m_s}$ (relative to the lexicographic order on words).

We have seen (cf. Proposition 1.2) that the rank of u has a binary representation ρ of length $\leq nH + O(\log n)$. The integers $m_1, ..., m_s$ are encoded by their binary representations $\mu_1, ..., \mu_s$ which all have length $\leq 1 + \lfloor \log n \rfloor$. Now, to encode $m_1, ..., m_s$ and the rank of u, we cannot just concatenate $\mu_1, ..., \mu_s, \rho$: how would we know where μ_1 stops, where μ_2 starts,..., in the word obtained by concatenation? Several tricks are possible to overcome the problem, they are described in §1.2.6. Using Proposition 1.5, we set $\xi = \langle \mu_1, ..., \mu_s, \rho \rangle$ which has length $|\xi| = |\rho| + O(|\mu_1| + ... + |\mu_s|) = nH + O(\log n)$ (Proposition 1.5 gives a much better bound but this is of no use here). Then u can be recovered from ξ which is a binary word of length $nH + O(\log n)$. Thus, asymptotically, we get a better upper bound than $n(H + 1)$, the one given by Shannon for prefix-free codes (cf. Theorem 1.3).

Of course, ξ is no more obtained from u via a morphism (i.e., a map which preserves concatenation of words) between the monoid of words in alphabet A and that of binary words.

Notice that this also shows that prefix-free codes are not the only way to efficiently encode into a binary word ξ a word u from alphabet $a_1, ..., a_s$ for which the numbers $m_1, ..., m_s$ of occurrences of the a_i's are known.

1.2.6 Coding finitely many words with one word

How can we code two words u, v with only one word? The simplest way is to consider $u\$v$ where \$ is a fresh symbol outside the alphabet of u and v. But what if we want to stick to binary words? As said above, the concatenation of u and v does not do the job: how can one recover the prefix u in uv? A simple trick is to also concatenate the length of $|u|$ in unary and delimitate it by a zero. Indeed, denoting by 1^p the word $\overbrace{1\dots1}^{p \text{ times}}$, one can recover u and v from the word $1^{|u|}0uv$: the length of the first block of 1's tells where to stop in the suffix uv to get u. In other words, the map $(u, v) \to 1^{|u|}0uv$ is injective from $\{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$. In this way, the code of the pair (u, v) has length $2|u| + |v| + 1$. This can obviously be extended to more arguments using the map $(u_1, ..., u_s, v) \mapsto 1^{|u_1|}0^{|u_2|} \dots \varepsilon^{|u_s|}\varepsilon' u_1 \dots u_s v$ (where $\varepsilon = 0$ is s is even and $\varepsilon = 1$ is s is odd and $\varepsilon' = 1 - \varepsilon$.

Proposition 1.5. *Let $s \geq 1$. There exists a map $\langle\ \rangle : (\{0,1\}^*)^{s+1} \to \{0,1\}^*$ which is injective and computable and such that, for all $u_1, ..., u_s, v \in \{0,1\}^*$, $|\langle u_1, ..., u_s, v\rangle| = 2(|u_1| + ... + |u_s|) + |v| + 1$.*

The following technical improvement will be needed in Part II §2.1.

Proposition 1.6. *There exists a map $\langle\ \rangle : (\{0,1\}^*)^{s+1} \to \{0,1\}^*$ which is injective and computable and such that, for all $u_1, ..., u_s, v \in \{0,1\}^*$,*

$$\begin{aligned} |\langle u_1, ..., u_s, v\rangle| &= (|u_1| + ... + |u_s|) + (\log |u_1| + ... + \log |u_s|) \\ &\quad +2(\log\log |u_1| + ... + \log\log |u_s|) + |v| + O(1) \end{aligned}$$

Proof. We consider the case $s = 1$, i.e., we want to code a pair (u, v). Instead of putting the prefix $1^{|u|}0$, let us put the binary representation $\beta(|u|)$ of the number $|u|$ prefixed by its length. This gives the more complex code: $1^{|\beta(|u|)|}0\beta(|u|)uv$ with length

$$|u| + |v| + 2(\lfloor \log |u| \rfloor + 1) + 1 \leq |u| + |v| + 2\log |u| + 3$$

The first block of ones gives the length of $\beta(|u|)$. Using this length, we can get $\beta(|u|)$ as the factor following this first block of ones. Now, $\beta(|u|)$ is the binary representation of $|u|$, so we get $|u|$ and can now separate u and v in the suffix uv. □

1.3 Probabilistic approach: ergodicity and lossy coding

The abstract probabilistic approach allows for considerable extensions of the results described in §1.2.

First, the restriction to fixed given frequencies can be relaxed. The probability of writing a_i may depend on what has already been written. For instance, Shannon's source coding theorem has been extended to the so called "ergodic asymptotically mean stationary source models".

Second, one can consider a lossy coding: some length n words in alphabet A are ill-treated or ignored. Let δ be the probability of this set of words. Shannon's theorem extends as follows:
- whatever close to 1 is $\delta < 1$, one can compress u only down to nH bits.
- whatever close to 0 is $\delta > 0$, one can achieve compression of u down to nH bits.

1.4 Algorithmic approach: Kolmogorov complexity

1.4.1 Berry's paradox

So far, we considered two kinds of binary codings for a word u in alphabet $a_1, ..., a_s$. The simplest one uses variable-length prefix-free codes (§1.2.2). The other one codes the rank of u as a member of some set (§1.2.5).
Clearly, there are plenty of other ways to encode any mathematical object. Why not consider all of them? And define the information content of a mathematical object x as *the shortest univoque description of x (written as a binary word).* Though quite appealing, this notion is ill defined as stressed by Berry's paradox[3]:

> Let N be the *lexicographically least binary word which cannot be univoquely described by any binary word of length less than* 1000.

This description of N contains 106 symbols of written English (including spaces) and, using ASCII codes, can be written as a binary word of length $106 \times 7 = 742$. Assuming such a description to be well defined would lead to a univoque description of N in 742 bits, hence less than 1000, a contradiction to the definition of N.
The solution to this inconsistency is clear: the quite vague notion of univoque description entering Berry's paradox is used both inside the sentence describing N and inside the argument to get the contradiction. A clash between two levels:
• the would be formal level carrying the description of N
• and the meta level which carries the inconsistency argument.
Any formalization of the notion of description should drastically reduce its scope and totally forbid any clash such as the above one.

1.4.2 The turn to computability

To get around the stumbling block of Berry's paradox and have a formal notion of description with wide scope, Andrei Nikolaievitch Kolmogorov (1903–1987) made an ingenious move: he turned to computability and replaced *description* by *computation program.* Exploiting the successful formalization of this a priori

[3] Berry's paradox is mentioned by Bertrand Russell in 1908 ([44], p.222 or 150), who credited G.G. Berry, an Oxford librarian, for the suggestion.

vague notion which was achieved in the thirties[4]. This approach was first announced by Kolmogorov in [24], 1963, and then developped in [25], 1965. Similar approaches were also independently developed by Solomonoff in [50], 1964, and by Chaitin in [6, 7], 1966-1969.

1.4.3 Digression on computability theory

The formalized notion of *computable function* (also called recursive function) goes along with that of *partial computable function* (also called partial recursive function) which should rather be called *partially computable partial function*, i.e., the *partial* character has to be distributed[5].
So, there are two theories :

- *the theory of computable functions*,
- *the theory of partial computable functions.*

The "right" theory, the one with a cornucopia of spectacular results, is that of partial computable functions.

> Let us pick up three fundamental results out of the cornucopia, which we state in terms of computers and programming languages. Let $\mathcal{I}$ and $\mathcal{O}$ be $\mathbb{N}$ or A^* where A is some finite or countably infinite alphabet (or, more generally, $\mathcal{I}$ and $\mathcal{O}$ can be elementary sets, cf. Definition 1.9).
>
> **Theorem 1.7.**
>
> 1. [Enumeration theorem] *The function which executes programs on their inputs:* (program, input) → output *is itself partial computable. Formally, this means that there exists a partial computable function*
>
> $$U : \{0,1\}^* \times \mathcal{I} \to \mathcal{O}$$
>
> *such that the family of partial computable function $\mathcal{I} \to \mathcal{O}$ is exactly $\{U_e \mid e \in \{0,1\}^*\}$ where $U_e(x) = U(e,x)$.*
> *Such a function U is called universal for partial computable functions $\mathcal{I} \to \mathcal{O}$.*
>
> 2. [Parameter theorem (or s^m_n thm)]. *One can exchange input and program* (this is von Neumann's key idea for computers).
> *Formally, this means that, letting $\mathcal{I} = \mathcal{I}_1 \times \mathcal{I}_2$, universal maps $U_{\mathcal{I}_1 \times \mathcal{I}_2}$ and $U_{\mathcal{I}_2}$ are such that there exists a computable total map $s : \{0,1\}^* \times \mathcal{I}_1 \to \{0,1\}^*$ such that, for all $e \in \{0,1\}^*$, $x_1 \in \mathcal{I}_1$ and $x_2 \in \mathcal{I}_2$,*
>
> $$U_{\mathcal{I}_1 \times \mathcal{I}_2}(e,(x_1,x_2)) = U_{\mathcal{I}_2}(s(e,x_1),x_2)$$

[4] Through the works of Alonzo Church (via lambda calculus), Alan Mathison Turing (via Turing machines) and Kurt Gödel and Jacques Herbrand (via Herbrand-Gödel systems of equations) and Stephen Cole Kleene (via the recursion and minimization operators).

[5] In French, Daniel Lacombe ([27], 1960) used the expression *semi-fonction semi-récursive.*

3. [Kleene fixed point theorem] *For any transformation of programs, there is a program which does the same* input → output *job as its transformed program*[6].
Formally, this means that, for every partial computable map $f : \{0,1\}^* \to \{0,1\}^*$, *there exists* e *such that*

$$\forall e \in \{0,1\}^* \quad \forall x \in \mathcal{I} \quad U(f(e), x) = U(e, x)$$

1.4.4 Kolmogorov complexity (or program size complexity)

Turning to computability, the basic idea for Kolmogorov complexity[7] can be summed up by the following equation:

$$\boxed{\textit{description} \quad = \quad \textit{program}}$$

When we say "program", we mean a program taken from a family of programs, i.e., written in a programming language or describing a Turing machine or a system of Herbrand-Gödel equations or a Post system,...
Since we are soon going to consider the length of programs, following what has been said in §1.1.2, we normalize programs: they will be binary words, i.e., elements of $\{0,1\}^*$.
So, we have to fix a function $\varphi : \{0,1\}^* \to \mathcal{O}$ and consider that the output of a program p is $\varphi(p)$.
Which φ are we to consider? Since we know that there are universal partial computable functions (i.e., functions able to emulate any other partial computable function modulo a computable transformation of programs, in other words, a compiler from one language to another), it is natural to consider universal partial computable functions. Which agrees with what has been said in §1.4.3.
Let us give the general definition of the Kolmogorov complexity associated to any function $\{0,1\}^* \to \mathcal{O}$.

Definition 1.8. *If* $\varphi : \{0,1\}^* \to \mathcal{O}$ *is a partial function, set* $K_\varphi : \mathcal{O} \to \mathbb{N}$

$$K_\varphi(y) = \min\{|p| : \varphi(p) = y\}$$

with the convention that $\min \emptyset = +\infty$.
Intuition: p *is a program (with no input),* φ *executes programs (i.e.,* φ *is altogether a programming language plus a compiler plus a machinery to run programs) and* $\varphi(p)$ *is the output of the run of program* p*. Thus, for* $y \in \mathcal{O}$, $K_\varphi(y)$ *is the length of shortest programs* p *with which* φ *computes* y *(i.e.,* $\varphi(p) = y$*).*

As said above, we shall consider this definition for partial computable functions $\{0,1\}^* \to \mathcal{O}$. Of course, this forces to consider a set $\mathcal{O}$ endowed with a computability structure. Hence the choice of sets that we shall call *elementary* which do not exhaust all possible ones but will suffice for the results mentioned in this paper.

[6] This is the seed of computer virology, cf. [4]

[7] Delahaye's books [11, 12] present a very attractive survey on Kolmogorov complexity.

Definition 1.9. *The family of elementary sets is obtained as follows:*
- it contains $\mathbb{N}$ and the A^*'s where A is a finite or countable alphabet,
- it is closed under finite (non empty) product, product with any non empty finite set and the finite sequence operator.

Note. Closure under the finite sequence operator is used to encode formulas in Theorem 2.4.

1.4.5 The invariance theorem

The problem with Definition 1.8 is that K_φ strongly depends on φ. Here comes a remarkable result, the invariance theorem, which insures that *there is a smallest K_φ, up to a constant.* It turns out that the proof of this theorem only needs the enumeration theorem and makes no use of the parameter theorem (usually omnipresent in computability theory).

Theorem 1.10 (Invariance theorem, Kolmogorov, [25], 1965)**.** *Let $\mathcal{O}$ be an elementary set (cf. Definition 1.9). Among the K_φ's, where $\varphi : \{0,1\}^* \to \mathcal{O}$ varies in the family $PC^{\mathcal{O}}$ of partial computable functions, there is a smallest one, up to an additive constant (= within some bounded interval). I.e.*

$$\exists V \in PC^{\mathcal{O}} \quad \forall \varphi \in PC^{\mathcal{O}} \quad \exists c \quad \forall y \in \mathcal{O} \quad K_V(y) \leq K_\varphi(y) + c$$

Such a V is called optimal.
Moreover, any universal partial computable function $\{0,1\}^* \to \mathcal{O}$ is optimal.

Proof. Let $U : \{0,1\}^* \times \{0,1\}^* \to \mathcal{O}$ be partial computable and universal for partial computable functions $\{0,1\}^* \to \mathcal{O}$ (cf. point 1 of Theorem 1.7).
Let $c : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be a total computable injective map such that $|c(e,x)| = 2|e| + |x| + 1$ (cf. Proposition 1.5).
Define $V : \{0,1\}^* \to \mathcal{O}$, with domain included in the range of c, as follows:

$$\forall e \in \{0,1\}^* \; \forall x \in \{0,1\}^* \quad V(c(e,x)) = U(e,x)$$

where equality means that both sides are simultaneously defined or not. Then, for every partial computable function $\varphi : \{0,1\}^* \to \mathcal{O}$, for every $y \in \mathcal{O}$, if $\varphi = U_e$ (i.e., $\varphi(x) = U(e,x)$ for all x, cf. point 1 of Theorem 1.7) then

$$\begin{aligned}
K_V(y) &= \text{least } |p| \text{ such that } V(p) = y \\
&\leq \text{least } |c(e,x)| \text{ such that } V(c(e,x)) = y \\
&\qquad\qquad \text{(least is relative to } x \text{ since } e \text{ is fixed)} \\
&= \text{least } |c(e,x)| \text{ such that } U(e,x)) = y \\
&= \text{least } |x| + 2|e| + 1 \text{ such that } \varphi(x) = y \\
&\qquad\qquad \text{since } |c(e,x)| = |x| + 2|e| + 1 \text{ and } \varphi(x) = U(e,x) \\
&= (\text{least } |x| \text{ such that } \varphi(x) = y) + 2|e| + 1 \\
&= K_\varphi(y) + 2|e| + 1
\end{aligned}$$

□

Using the invariance theorem, the Kolmogorov complexity $K^{\mathcal{O}} : \mathcal{O} \to \mathbb{N}$ is defined as K_V where V is any fixed optimal function. The arbitrariness of the choice of V does not modify drastically K_V, merely up to a constant.

Definition 1.11. *Kolmogorov complexity $K^{\mathcal{O}} : \mathcal{O} \to \mathbb{N}$ is K_V, where V is some fixed optimal partial function $\{0,1\}^* \to \mathcal{O}$. When $\mathcal{O}$ is clear from context, we shall simply write K.*
$K^{\mathcal{O}}$ is therefore minimum among the K_φ's, up to an additive constant.
$K^{\mathcal{O}}$ is defined up to an additive constant: if V and V' are both optimal then

$$\exists c \quad \forall x \in \mathcal{O} \quad |K_V(x) - K_{V'}(x)| \leq c$$

1.4.6 What Kolmogorov said about the constant

So Kolmogorov complexity is an integer defined up to a constant...! But the constant is uniformly bounded for $x \in \mathcal{O}$.
Let us quote what Kolmogorov said about the constant in [25], 1965:

> *Of course, one can avoid the indeterminacies associated with the [above] constants, by considering particular [...functions V], but it is doubtful that this can be done without explicit arbitrariness.*
> *One must, however, suppose that the different "reasonable" [above optimal functions] will lead to "complexity estimates" that will converge on hundreds of bits instead of tens of thousands.*
> *Hence, such quantities as the "complexity" of the text of "War and Peace" can be assumed to be defined with what amounts to uniqueness.*

In fact, this constant witnesses the multitude of models of computation: universal Turing machines, universal cellular automata, Herbrand-Gödel systems of equations, Post systems, Kleene definitions,... If we feel that one of them is canonical then we may consider the associated Kolmogorov complexity as the right one and forget about the constant. This has been developed for Schoenfinkel-Curry combinators S, K, I by Tromp, cf. [31] §3.2.2–3.2.6.
However, even if we fix a particular K_V, the importance of the invariance theorem remains since it tells us that K is less than *any* K_φ (up to a constant). A result which is applied again and again to develop the theory.

1.4.7 Considering inputs: conditional Kolmogorov complexity

In the enumeration theorem, we considered *(program, input) → output* functions (cf. Theorem 1.7). Then, in the definition of Kolmogorov complexity, we gave up the inputs, dealing with *program → output* functions.
Conditional Kolmogorov complexity deals with the inputs. Instead of measuring the information content of $y \in \mathcal{O}$, we measure it given as free some object z, which may help to compute y. A trivial case is when $z = y$, then the information content of y given y is null. In fact, there is an obvious program which outputs exactly its input, whatever be the input.

Let us mention that, in computer science, inputs are also considered as *environments.*
Let us state the formal definition and the adequate invariance theorem.

Definition 1.12. *If $\varphi : \{0,1\}^* \times \mathcal{I} \to \mathcal{O}$ is a partial function, set $K_\varphi(\ \mid\ ) : \mathcal{O} \times \mathcal{I} \to \mathbb{N}$*

$$K_\varphi(y \mid z) = \min\{|p| \mid \varphi(p,z) = y\}$$

Intuition: p is a program (with expects an input z), φ executes programs (i.e., φ is altogether a programming language plus a compiler plus a machinery to run programs) and $\varphi(p,z)$ is the output of the run of program p on input z. Thus, for $y \in \mathcal{O}$, $K_\varphi(y \mid z)$ is the length of shortest programs p with which φ computes y on input z (i.e., $\varphi(p,z) = y$).

Theorem 1.13 (Invariance theorem for conditional complexity)**.** *Among the $K_\varphi(\ \mid\ )$'s, where φ varies in the family $PC^{\mathcal{O}}_{\mathcal{I}}$ of partial computable functions $\{0,1\}^* \times \mathcal{I} \to \mathcal{O}$, there is a smallest one, up to an additive constant (i.e., within some bounded interval) :*

$$\exists V \in PC^{\mathcal{O}}_{\mathcal{I}} \quad \forall \varphi \in PC^{\mathcal{O}}_{\mathcal{I}} \quad \exists c \quad \forall y \in \mathcal{O} \quad \forall z \in \mathcal{I} \quad K_V(y \mid z) \leq K_\varphi(y \mid z) + c$$

Such a V is called optimal.
Moreover, any universal partial computable map $\{0,1\}^* \times \mathcal{I} \to \mathcal{O}$ is optimal.

The proof is similar to that of Theorem 1.10.

Definition 1.14. *$K^{\mathcal{I} \to \mathcal{O}} : \mathcal{O} \times \mathcal{I} \to \mathbb{N}$ is $K_V(\ \mid\ )$ where V is some fixed optimal partial function.*

$K^{\mathcal{I} \to \mathcal{O}}$ is defined up to an additive constant: if V et V' are both optimal then

$$\exists c \quad \forall y \in \mathcal{O} \quad \forall z \in \mathcal{I} \quad |K_V(y \mid z) - K_{V'}(y \mid z)| \leq c$$

Again, an integer defined up to a constant...! However, the constant is uniform in $y \in \mathcal{O}$ and $z \in \mathcal{I}$.

1.4.8 Simple upper bounds for Kolmogorov complexity

Finally, let us mention rather trivial upper bounds:
- the information content of a word is at most its length.
- conditional complexity cannot be harder than the non conditional one.

Proposition 1.15.

1. *There exists c such that*

$$\forall x \in \{0,1\}^* \quad K^{\{0,1\}^*}(x) \leq |x| + c \quad , \quad \forall n \in \mathbb{N} \quad K^{\mathbb{N}}(n) \leq \log(n) + c$$

2. *There exists c such that*

$$\forall x \in \mathcal{O} \quad \forall y \in \mathcal{I} \quad K^{\mathcal{I} \to \mathcal{O}}(x \mid y) \leq K^{\mathcal{O}}(x) + c$$

3. *Let $f : \mathcal{O} \to \mathcal{O}'$ be computable. There exists c such that*

$$\begin{array}{llrcl} \forall x \in \mathcal{O} & & K^{\mathcal{O}'}(f(x)) & \leq & K^{\mathcal{O}}(x) + c \\ \forall x \in \mathcal{O} & \forall Y \in \mathcal{I} & K^{\mathcal{I}\to\mathcal{O}'}(f(x) \mid y) & \leq & K^{\mathcal{I}\to\mathcal{O}}(x \mid y) + c \end{array}$$

Proof. We only prove 1. Let $Id : \{0,1\}^* \to \{0,1\}^*$ be the identity function. The invariance theorem insures that there exists c such that $K^{\{0,1\}^*} \leq K_{Id}^{\{0,1\}^*} + c$. Now, it is easy to see that $K_{Id}^{\{0,1\}^*} = |x|$, so that $K^{\{0,1\}^*}(x) \leq |x| + c$.
Let $\theta : \{0,1\}^* \to \mathbb{N}$ be the function (which is, in fact, a bijection) which associates to a word $u = a_{k-1}...a_0$ the integer

$$\theta(u) = (2^k + a_{k-1}2^{k-1} + ... + 2a_1 + a_0) - 1$$

(i.e., the predecessor of the integer with binary representation $1u$). Clearly, $K_\theta^{\mathbb{N}}(n) = \lfloor \log(n+1) \rfloor$. The invariance theorem insures that there exists c such that $K^{\mathbb{N}} \leq K_\theta^{\mathbb{N}} + c$. Hence $K^{\mathbb{N}}(n) \leq \log(n) + c + 1$ for all $n \in \mathbb{N}$. □

The following technical property is a variation of an argument already used in §1.2.5: the rank of an element in a set defines this element, and if the set is computable, so is this process.

Proposition 1.16. *Let $A \subseteq \mathbb{N} \times \mathcal{O}$ be computable such that $A_n = A \cap (\{n\} \times \mathcal{O})$ is finite for all n. Then, letting $\sharp X$ be the number of elements of X,*

$$\exists c \quad \forall x \in A_n \quad K(x \mid n) \leq \log(\sharp(A_n)) + c$$

Proof. Observe that x is determined by its rank in A_n. This rank is an integer $< \sharp A_n$ hence its binary representation has length $\leq \lfloor \log(\sharp A_n) \rfloor + 1$. □

2 Kolmogorov complexity and undecidability

2.1 K is unbounded

Let $K = K_V : \mathcal{O} \to \mathbb{N}$ where $V : \{0,1\}^* \to \mathcal{O}$ is optimal (cf. Theorem §1.10). Since there are finitely many programs of size $\leq n$ (namely, the $2^{n+1} - 1$ binary words of size $\leq n$), there are finitely many elements of $\mathcal{O}$ with Kolmogorov complexity less than n. This shows that K is unbounded.

2.2 K is not computable

Berry's paradox (cf. §1.4.1) has a counterpart in terms of Kolmogorov complexity: it gives a very simple proof that K, which is a total function $\mathcal{O} \to \mathbb{N}$, is not computable.

Proof that K is not computable. For simplicity of notations, we consider the case $\mathcal{O} = \mathbb{N}$. Define $L : \mathbb{N} \to \mathcal{O}$ as follows:

$$L(n) \quad = \quad \text{least } k \text{ such that } K(k) \geq 2n$$

So that $K(L(n)) \geq 2n$ for all n. If K were computable so would be L. Let $V : \mathcal{O} \to \mathbb{N}$ be optimal, i.e., $K = K_V$. The invariance theorem insures that there exists c such that $K \leq K_L + c$. Observe that $K_L(L(n) \leq n$ by definition of K_L. Thus,

$$2n \leq K(L(n)) \leq K_L(L(n) + c \leq n + c$$

A contradiction for $n > c$. □

The non computability of K can be seen as a version of the undecidability of the halting problem. In fact, there is a simple way to compute K when the halting problem is used as an oracle. To get the value of $K(x)$, proceed as follows:

- enumerate the programs in $\{0,1\}^*$ in lexicographic order,
- for each program p, check if $V(p)$ halts (using the oracle),
- in case $V(p)$ halts then compute its value,
- halt and output $|p|$ when some p is obtained such that $V(p) = x$.

The converse is also true: one can prove that *the halting problem is computable with K as an oracle.*

The argument for the undecidability of K can be used to prove a much stronger statement: K can not be bounded from below by any unbounded partial computable function.

Theorem 2.1 (Kolmogorov)**.** *There is no unbounded partial recursive function $\psi : \mathcal{O} \to \mathbb{N}$ such that $\psi(x) \leq K(x)$ for all x in the domain of ψ.*

Of course, K is bounded from above by a total computable function, cf. Proposition 1.15.

2.3 K is computable from above

Though K is not computable, it can be approximated from above. The idea is simple. Suppose $\mathcal{O} = \{0,1\}^*$. Let c be as in point 1 of Proposition 1.15. Consider all programs of length less than $|x| + c$ and let them be executed during t steps. If none of them converges and outputs x then take $|x| + c$ as a t-bound. If some of them converges and outputs x then the bound is the length of the shortest such program.
The limit of this process is $K(x)$, it is obtained at some finite step which we are not able to bound.
Formally, this means that there is some $F : \mathcal{O} \times \mathbb{N} \to \mathbb{N}$ which is computable and decreasing in its second argument such that

$$K(x) = \lim_{t \to +\infty} F(x,t) = \min\{F(x,t) \mid t \in \mathbb{N}\}$$

2.4 Kolmogorov complexity and Gödel's incompleteness theorem

A striking version of Gödel's incompleteness theorem has been given by Chaitin in [8, 9], 1971-1974, in terms of Kolmogorov complexity. Since Gödel's celebrated proof of the incompleteness theorem, we know that, in the language of arithmetic, one can formalize computability and logic. In particular, one can formalize Kolmogorov complexity and statements about it. Chaitin's proves a version of the incompleteness theorem which insures that among true unprovable formulas there are all true statements $K(u) > n$ for n large enough.

Theorem 2.2 (Chaitin, [9], 1974)**.** *Let $\mathcal{T}$ be a computably enumerable set of axioms in the language of arithmetic. Suppose that all axioms in $\mathcal{T}$ are true in the standard model of arithmetics with base $\mathbb{N}$. Then there exists N such that if $\mathcal{T}$ proves $K(u) > n$ (with $u \in \{0,1\}^*$ and $n \in \mathbb{N}$) then $n \leq N$.*

How the constant N depends on $\mathcal{T}$ has been giving a remarkable analysis by Chaitin. To that purpose, he extends Kolmogorov complexity to computably enumerable sets.

Definition 2.3 (Chaitin, [9], 1974)**.** *Let $\mathcal{O}$ be an elementary set (cf. Definition 1.9) and $\mathcal{CE}$ be the family of computably enumerable (c.e.) subsets of $\mathcal{O}$. To any partial computable $\varphi : \{0,1\}^* \times \mathbb{N} \to \mathcal{O}$, associate the Kolmogorov complexity $K_\varphi : \mathcal{CE} \to \mathbb{N}$ such that, for all c.e. subset $\mathcal{T}$ of $\mathcal{O}$,*

$$K_\varphi(\mathcal{T}) = \min\{|p| \mid \mathcal{T} = \{\varphi(p,t) \mid t \in \mathbb{N}\}\}$$

(observe that $\{\varphi(p,t) \mid t \in \mathbb{N}\}$ is always c.e. and any c.e. subset of $\mathcal{O}$ can be obtained in this way for some φ).

The invariance theorem still holds for this notion of Kolmogorov complexity, leading to the following notion.

Definition 2.4 (Chaitin, [9], 1974)**.** *$K^{\mathcal{CE}} : \mathcal{CE} \to \mathbb{N}$ is K_φ where φ is some fixed optimal partial function. It is defined up to an additive constant.*

We can now state how the constant N in Theorem 2.2 depends on the theory $\mathcal{T}$.

Theorem 2.5 (Chaitin, [9], 1974)**.** *There exists a constant c such that, for all c.e. sets $\mathcal{T}$ satisfying the hypothesis of Theorem 2.2, the associated constant N is such that*

$$N \leq K^{\mathcal{CE}}(\mathcal{T}) + c$$

Chaitin also reformulates Theorem 2.2 as follows:

> *If $\mathcal{T}$ consist of true formulas then it cannot prove that a string has Kolmogorov complexity greater than the Kolmogorov complexity of $\mathcal{T}$ itself (up to a constant independent of $\mathcal{T}$).*

Remark. 2.6. The previous statement, and Chaitin's assertion that the Kolmogorov complexity of $\mathcal{T}$ somehow measures the power of $\mathcal{T}$ as a theory, has been much criticized in van Lambalgen ([28], 1989), Fallis ([14], 1996) and Raatikainen ([43], 1998). Raatikainen's main argument in [43] against Chaitin's interpretation is that the constant in Theorem 2.2 strongly depends on the choice of the optimal function V such that $K = K_V$. Indeed, for any fixed

theory $\mathcal{T}$, one can choose such a V so that the constant is zero! And also choose V so that the constant is arbitrarily large.
Though these arguments are perfectly sound, we disagree with the criticisms issued from them. Let us detail three main rebuttals.

- First, such arguments are based on the use of optimal functions associated to very unnatural universal functions V (cf. point 1 of Theorem 1.7 and the last assertion of Theorem 1.10). It has since been recognized that universality is not always sufficient to get smooth results. Universality by prefix adjunction is sometimes required, (cf., for instance, §2.1 and §6 in Becher, Figueira, Grigorieff & Miller, 2006). This means that, for an enumeration $(\varphi_e)_{e\in\{0,1\}^*}$ of partial computable functions, the optimal function V is to satisfy equality $V(ep) = \varphi_e(p)$, for all e, p, where ep is the concatenation of the strings e and p.

- Second, and more important than the above technical counterargument, it is a simple fact that modelization rarely rules out all pathological cases. It is intended to be used in "reasonable" cases. Of course, this may be misleading, but perfect modelization is illusory. In our opinion, this is best illustrated by Kolmogorov's citation quoted in §1.4.6 to which Raatikainen's argument could be applied mutatis mutandis: there are optimal functions for which the complexity of the text of "War and Peace" is null and other ones for which it is arbitrarily large. Nevertheless, this does not prevent Kolmogorov to assert (in the founding paper of the theory [25]): *[For] "reasonable" [above optimal functions], such quantities as the "complexity" of the text of "War and Peace" can be assumed to be defined with what amounts to uniqueness.*

- Third, a final technical answer to such criticisms has been recently provided by Calude & Jurgensen in [5], 2005. They improve the incompleteness result given by Theorem 2.2, proving that, for a class of formulas in the vein of those in that theorem, the probability that such a formula of length n is provable tends to zero when n tends to infinity whereas the probability that it be true has a strictly positive lower bound.

3 Kolmogorov complexity: some variations

Note. The denotations of (plain) Kolmogorov complexity (that of §1.4.5) and its prefix version (cf. 3.3) may cause some confusion. They long used to be respectively denoted by K and H in the literature. But in their book [31] (first edition, 1993), Li & Vitanyi respectively denoted them by C and K. Due to the large success of this book, these last denotations are since used in many papers. So that two incompatible denotations now appear in the literature. In this paper, we stick to the traditional denotations K and H.

3.1 Levin monotone complexity

Kolmogorov complexity is non monotone, be it on $\mathbb{N}$ with the natural ordering or on $\{0,1\}^*$ with the lexicographic ordering. In fact, for every n and c, there are strings of length n with complexity $\geq n(1-2^{-c})$ (cf. Proposition 4.2). However, since $n \mapsto 1^n$ is computable, $K(1^n) \leq K(n) + O(1) \leq \log n + O(1)$ (cf. point 3 of Proposition 1.15) is much less than $n(1-2^{-c})$ for n large enough.

Leonid Levin ([29], 1973) introduced a monotone version of Kolmogorov complexity. The idea is to consider possibly infinite computations of Turing machines which never erase anything on the output tape. Such machines have finite or infinite outputs and compute total maps $\{0,1\}^* \to \{0,1\}^{\leq\omega}$ where $\{0,1\}^{\leq\omega} = \{0,1\}^* \cup \{0,1\}^{\mathbb{N}}$ is the family of finite or infinite binary strings. These maps can also be viewed as limit maps $p \to \sup_{t\to\infty} \varphi(p,t)$ where $\varphi : \{0,1\}^* \times \mathbb{N} \to \{0,1\}^*$ is total monotone non decreasing in its second argument.
To each such map φ, Levin associates a monotone non decreasing map $K_\varphi^{mon} : \{0,1\}^* \to \mathbb{N}$ such that

$$K_\varphi^{mon}(x) = \min\{|p| \mid \exists t\ x \leq_{pref} \varphi(p,t)\}$$

Theorem 3.1 (Levin ([29], 1973).

1. If φ is total computable and monotone non decreasing in its second argument then $K_\varphi^{mon} : \{0,1\}^* \to \mathbb{N}$ is monotone non decreasing:

$$x \leq_{pref} y \Rightarrow K_\varphi^{mon}(x) \leq K_\varphi^{mon}(y)$$

2. Among the K_φ^{mon}'s, φ total computable monotone non decreasing in its second argument, there exists a smallest one, up to a constant.

Considering total φ's in the above theorem is a priori surprising since there is no computable enumeration of total computable functions and the proof of the Invariance Theorem 1.10 is based on the enumeration theorem (cf. Theorem 1.7). The trick to overcome that problem is as follows.

- Consider all partial computable $\varphi : \{0,1\}^* \times \mathbb{N} \to \{0,1\}^*$ which are total monotone non decreasing in their second argument.
- Associate to each such φ a total $\widetilde{\varphi}$ defined as follows: $\widetilde{\varphi}(p,t)$ is the largest $\varphi(p,t')$ such that $t' \leq t$ and $\varphi(t')$ is defined within $t+1$ computation steps if there is such a t'. If there is none then $\widetilde{\varphi}(p,t)$ is the empty word.
- Observe that $K_\varphi^{mon}(x) = K_{\widetilde{\varphi}}^{mon}(x)$.

In §5.2.3, we shall see some remarkable property of Levin monotone complexity K^{mon} concerning Martin-Löf random reals.

3.2 Schnorr process complexity

Another variant of Kolmogorov complexity has been introduced by Klaus Peter Schnorr in [47], 1973. It is based on the subclass of partial computable functions $\varphi : \{0,1\}^* \to \{0,1\}^*$ which are monotone non decreasing relative to the prefix ordering:

$$(*) \qquad (p \leq_{pref} q \ \wedge \ \varphi(p), \varphi(q) \text{ are both defined}) \ \Rightarrow \ \varphi(p) \leq_{pref} \varphi(q)$$

Why such a requirement on φ? The reason can be explained as follows.

- Consider a sequential composition (i.e., a pipeline) of two processes, formalized as two functions f, g. The first one takes an input p and outputs $f(p)$, the second one takes $f(p)$ as input and outputs $g(f(p))$.
- Each process is supposed to be monotone: the first letter of $f(p)$ appears first, then the second one, etc. Idem with the digits of $g(q)$ for any input q.
- More efficiency is obtained if one can develop the computation of g on input $f(p)$ as soon as the letters of $f(p)$ appear. More precisely, suppose the prefix q of $f(p)$ has already appeared but there is some delay to get the subsequent letters. Then we can compute $g(q)$. But this is useful only in case the computation of $g(q)$ is itself a prefix of that of $g(f(p))$. This last condition is exactly the requirement $(*)$.

An enumeration theorem holds for the φ's satisfying $(*)$, allowing to prove an invariance theorem and to define a so-called process complexity $K^{proc} : \{0,1\}^* \to \mathbb{N}$. The same remarkable property of Levin's monotone complexity also holds with Schnorr process complexity, cf. §5.2.3.

3.3 Prefix (or self-delimited) complexity

Levin ([30], 1974), Gács ([18], 1974) and Chaitin ([10], 1975) introduced the most successful variant of Kolmogorov complexity: the prefix complexity. The idea is to restrict the family of partial computable functions $\{0,1\}^* \to \mathcal{O}$ (recall $\mathcal{O}$ denotes an elementary set in the sense of Definition 1.9) to those which have prefix-free domains, i.e. any two words in the domain are incomparable with respect to the prefix ordering.

An enumeration theorem holds for the φ's satisfying $(*)$, allowing to prove an invariance theorem and to define the so-called prefix complexity $H : \{0,1\}^* \to \mathbb{N}$ (not to be confused with the entropy of a family of frequencies, cf. §1.2.3).

Theorem 3.2. *Among the K_φ's, where $\varphi : \{0,1\}^* \to \mathcal{O}$ varies over partial computable functions with prefix-free domain, there exists a smallest one, up to a constant. This smallest one (defined up to a constant), denoted by $H^{\mathcal{O}}$, is called the prefix complexity.*

This prefix-free condition on the domain may seem rather technical. A conceptual meaning of this condition has been given by Chaitin in terms of self-delimitation.

Proposition 3.3 (Chaitin, [10], 1975)**.** *A partial computable function $\varphi : \{0,1\}^* \to \mathcal{O}$ has prefix-free domain if and only if it can be computed by a Turing machine $\mathcal{M}$ with the following property:*

> *If x is in domain(φ) (i.e., $\mathcal{M}$ on input p halts in an accepting state at some step) then the head of the input tape of $\mathcal{M}$ reads entirely the input p but never moves to the cell right to p.*

This means that p, interpreted as a program, has no need of external action (as that of an end-of-file symbol) to know where it ends: as Chaitin says, the program is self-delimited. A comparison can be made with biological phenomena. For instance, the hand of a person grows during its childhood and then stops growing. No external action prevents the hand to go on growing. There is something inside the genetic program which creates a halting signal so that the hand stops growing.

The main reason for the success of the prefix complexity is that, with prefix-free domains, one can use the Kraft-Chaitin inequality (cf. the proof of Theorem 1.3 in §1.2.4) and get remarkable properties.

Theorem 3.4 (Kraft-Chaitin inequality)**.** *A sequence (resp. computable sequence) $(n_i)_{i\in\mathbb{N}}$ of non negative integers is the sequence of lengths of a prefix-free (resp. computable) family of words $(u_i)_{i\in\mathbb{N}}$ if and only if $\sum_{i\in\mathbb{N}} 2^{-n_i} \leq 1$.*

Let us state the most spectacular property of the prefix complexity.

Theorem 3.5 (The Coding Theorem (Levin ([30], 1974))**.** *Consider the family $\ell_1^{\text{c.e.}}$ of sequences of non negative real numbers $(r_x)_{x\in\mathcal{O}}$ such that*

- *$\sum_{x\in\mathcal{O}} r_x < +\infty$ (i.e., the series is summable),*
- *$\{(x,q) \in \mathcal{O}\times\mathbb{Q} \mid q < r_x\}$ is computably enumerable (i.e., the r_x's have c.e. left cuts in the set of rational numbers $\mathbb{Q}$ and this is uniform in x).*

The sequence $(2^{-H^{\mathcal{O}}(x)})_{x\in\mathcal{O}}$ is in $\ell_1^{\text{c.e.}}$ and, up to a multiplicative factor, it is the largest sequence in $\ell_1^{\text{c.e.}}$. This means that

$$\forall (r_x)_{x\in\mathcal{O}} \in \ell_1^{\text{c.e.}} \quad \exists c \quad \forall x \in \mathcal{O} \quad r_x \leq c\, 2^{-H^{\mathcal{O}}(x)}$$

In particular, consider a countably infinite alphabet A. Let $V : \{0,1\}^* \to A$ be a partial computable function with prefix-free domain such that $H^A = K_V$. Consider the prefix code $(p_a)_{a\in A}$ such that, for each letter $a \in A$, p_a is a shortest binary string such that $V(p_a) = a$. Then, for every probability distribution $P : A \to [0,1]$ over the letters of the alphabet A, which is computably approximable from below (i.e., $\{(a,q) \in A\times\mathbb{Q} \mid q < P(a)\}$ is computably enumerable), we have

$$\forall a \in A \quad P(a) \leq c\, 2^{-H^A(a)}$$

for some c which depends on P but not on $a \in A$. This inequality is the reason why the sequence $(2^{-H^A(a)})_{a \in A}$ is also called *the universal a priori probability* (though, strictly speaking, it is not a probablity since the $2^{-H^A(a)}$'s do not sum up to 1).

3.4 Oracular Kolmogorov complexity

As is always the case in computability theory, everything relativizes to any oracle Z. Relativization modifies the equation given at the start of §1.4.4, which is now

$$\textit{description} \quad = \quad \textit{program of a partial } Z\textit{-computable function}$$

and for each possible oracle Z there exists a Kolmogorov complexity relative to oracle Z.

Oracles in computability theory can also be considered as second-order arguments of computable or partial computable *functionals*. The same holds with oracular Kolmogorov complexity: the oracle Z can be seen as a second-order condition for a *second-order conditional Kolmogorov complexity*

$$K(y \mid Z) \quad \text{where} \quad K(\ \mid\ ) : \mathcal{O} \times P(\mathcal{I}) \to \mathbb{N}$$

Which has the advantage that the unavoidable constant in the "up to a constant" properties does not depend on the particular oracle. It depends solely on the considered functional.
Finally, one can mix first-order and second-order conditions, leading to a conditional Kolmogorov complexity with both first-order and second-order conditions

$$K(y \mid z, Z) \quad \text{where} \quad K(\ \mid\ ,\ ) : \mathcal{O} \times \mathcal{I} \times P(\mathcal{I}) \to \mathbb{N}$$

We shall see in §5.6.2 an interesting property involving oracular Kolmogorov complexity.

3.5 Sub-oracular Kolmogorov complexity

Going back to the idea of possibly infinite computations as in §3.1, Let us define $K^\infty : \{0,1\}^* \to \mathbb{N}$ such that

$$K^\infty(x) = \min\{|p| \mid U(p) = x\}$$

where U is the map $\{0,1\}^* \to \{0,1\}^{\leq \omega}$ computed by a universal Turing machine with possibly infinite computations. This complexity lies between K and $K(\ \mid \emptyset')$ (where $\emptyset'$ is a computably enumerable set which encodes the halting problem):

$$\forall x \quad K(x \mid \emptyset') \leq K^\infty(x) + O(1) \leq K(x) + O(1)$$

This complexity is studied in [1], 2005, by Becher, Figueira, Nies & Picci, and also in our paper [17], 2006.

4 Formalization of randomness: finite objects

4.1 Sciences of randomness: probability theory

Random objects *(words, integers, reals,...)* constitute the basic intuition for probabilities ... *but they are not considered per se.* No formal definition of random object is given: there seems to be no need for such a formal concept. The existing formal notion of *random variable* has nothing to do with randomness: a random variable is merely a *measurable function* which can be as non random as one likes.

It sounds strange that the mathematical theory which deals with randomness removes the natural basic questions:

- *What is a random string?*
- *What is a random infinite sequence?*

When questioned, people in probability theory agree that they skip these questions but do not feel sorry about it. As it is, the theory deals with laws of randomness and is so successful that it can do without entering this problem.

This may seem to be analogous to what is the case in geometry. What are points, lines, planes? No definition is given, only relations between them. Giving up the quest for an analysis of the nature of geometrical objects in profit of the axiomatic method has been a considerable scientific step.
However, we contest such an analogy. Random objects are heavily used in many areas of science and technology: sampling, cryptology,... Of course, such objects are in fact *"as much as we can random"*. Which means *fake randomness.* But they refer to an ideal notion of randomness which cannot be simply disregarded.

In fact, since Pierre Simon de Laplace (1749–1827), some probabilists never gave up the idea of formalizing the notion of random object. Let us cite particularly Richard von Mises (1883–1953) and Kolmogorov. In fact, it is quite impressive that, having so brilliantly and efficiently axiomatized probability theory via measure theory in [23], 1933, Kolmogorov was not fully satisfied of such foundations[8]. And he kept a keen interest to the quest for a formal notion of randomness initiated by von Mises in the 20's.

4.2 The 100 heads paradoxical result in probability theory

That probability theory fails to completely account for randomness is strongly witnessed by the following paradoxical fact. In probability theory, *if we toss an unbiaised coin 100 times then 100 heads are just as probable as any other outcome!* Who really believes that?

> *The axioms of probability theory, as developped by Kolmogorov, do not solve all mysteries that they are sometimes supposed to.*

[8] Kolmogorov is one of the rare probabilists – up to now – not to believe that Kolmogorov's axioms for probability theory do not constitute the last word about formalizing randomness...

Gács, [20], 1993

4.3 Sciences of randomness: cryptology

Contrarily to probability theory, cryptology heavily uses random objects. Though again, no formal definition is given, random sequences are produced which are not fully random, just hard enough so that the mechanism which produces them cannot be discovered in reasonable time.

> *Anyone who considers arithmetical methods of producing random reals is, of course, in a state of sin. For, as has been pointed out several times, there is no such thing as a random number — there are only methods to produce random numbers, and a strict arithmetical procedure is of course not such a method.*
>
> Von Neumann, [40], 1951

So, what is "true" randomness? Is there something like a degree of randomness? Presently, (fake) randomness only means to pass some statistical tests. One can ask for more.

4.4 Kolmogorov's proposal: incompressible strings

We now assume that $\mathcal{O} = \{0,1\}^*$, i.e., we restrict to words.

4.4.1 Incompressibility with Kolmogorov complexity

Though much work had been devoted to get *a mathematical theory of random objects*, notably by von Mises ([35, 36], 1919-1939), none was satisfactory up to the 60's when Kolmogorov based such a theory on Kolmogorov complexity, hence on computability theory.
The theory was, in fact, independently[9] developed by Gregory J. Chaitin (b. 1947), [6, 7] who submitted both papers in 1965.

The basic idea is as follows:

- *larger is the Kolmogorov complexity of a text, more random is the text,*
- *larger is its information content, and more compressed is the text.*

Thus, a theory for measuring the information content is also a theory of randomness.

Recall that there exists c such that for all $x \in \{0,1\}^*$, $K(x) \leq |x| + c$ (Proposition 1.15). The reason being that there is a "stupid" program of length about $|x|$ which computes the word x by telling what are the successive letters of x.
The intuition of incompressibility is as follows: x is incompressible if there no shorter way to get x.
Of course, we are not going to define absolute randomness for words. But a measure of randomness telling *how far from* $|x|$ *is* $K(x)$.

[9] For a detailed analysis of *who did what, and when*, see Li & Vitanyi's book [31], p.89–92.

Definition 4.1 (Measure of incompressibility).
A word x is c-incompressible if $K(x) \geq |x| - c$.

It is rather intuitive that most things are random. The next Proposition formalizes this idea.

Proposition 4.2. *For any n, the proportion of c-incompressible strings of length n is $\geq 1 - 2^{-c}$.*

> *Proof.* At most $2^{n-c} - 1$ programs of length $< n - c$ and 2^n strings of length n. □

4.4.2 Incompressibility with length conditional Kolmogorov complexity

We observed in §1.2.3 that the entropy of a word of the form 000...0 is null. i.e., entropy did not considered the information conveyed by the length.
Here, with incompressibility based on Kolmogorov complexity, we can also ignore the information content conveyed by the length by considering *incompressibility based on length conditional Kolmogorov complexity.*

Definition 4.3 (Measure of length conditional incompressibility). *A word x is length conditional c-incompressible if $K(x \mid |x|) \geq |x| - c$.*

The same simple counting argument yields the following Proposition.

Proposition 4.4. *For all n, the proportion of length conditional c-incompressible strings of length n is $\geq 1 - 2^{-c}$.*

A priori length conditional incompressibility is stronger than mere incompressibility. However, the two notions of incompressibility are about the same ... up to a constant.

Proposition 4.5. *There exists d such that, for all $c \in \mathbb{N}$ and $x \in \{0,1\}^*$*

1. *x is length conditional c-incompressible $\Rightarrow$ x is $(c+d)$-incompressible*
2. *x is c-incompressible $\Rightarrow$ x is length conditional $(2c+d)$-incompressible.*

Proof. 1 is trivial. For 2, first observe that there exists e such that, for all x,

$$(*) \quad K(x) \leq K(x \mid |x|) + 2K(|x| - K(x \mid |x|)) + d$$

In fact, if $K = K_\varphi$ and $K(\ \mid\ ) = K_{\psi(\ \mid\ )}$, consider p, q such that

$$\begin{array}{rclcrcl} |x| - K(x \mid |x|) & = & \varphi(p) & \quad & \psi(q \mid |x|) & = & x \\ K(|x| - K(x \mid |x|)) & = & |p| & & K(x \mid |x|) & = & |q| \end{array}$$

With p and q, hence with $\langle p, q\rangle$ (cf. Proposition 1.5), one can successively get $\left\{ \begin{array}{ll} |x| - K(x \mid |x|) & \text{this is } \varphi(p) \\ K(x \mid |x|) & \text{this is } q \\ |x| & \text{just sum the above quantities} \\ x & \text{this is } \psi(q \mid |x|) \end{array} \right.$

Thus, $K(x) \leq |\langle p, q\rangle| + O(1)$. Applying Proposition 1.5, we get (*). Using $K^{\mathbb{N}} \leq \log + c_1$ and $K^{\{0,1\}^*}(x) \geq |x| - c$ (cf., Proposition 1.15), (*) yields

$$|x| - K(x \mid |x|) \leq 2\log(|x| - K(x \mid |x|)) + 2c_1 + c + d$$

Finally, observe that $z \leq 2\log z + k$ insures $z \leq \max(8, 2k)$. □

4.5 Incompressibility is randomness: Martin-Löf's argument

Now, if incompressibility is clearly a necessary condition for randomness, how do we argue that it is a sufficient condition? Contraposing the wanted implication, let us see that if a word fails some statistical test then it is not incompressible. We consider some spectacular failures of statistical tests.

Example 4.6.

1. [Constant half length prefix] *For all n large enough, a string $0^n u$ with $|u| = n$ cannot be c-incompressible.*

2. [Palindromes] *Large enough palindromes cannot be c-incompressible.*

3. [0 and 1 not equidistributed] *For all $0 < \alpha < 1$, for all n large enough, a string of length n which has $\leq \alpha\frac{n}{2}$ zeros cannot be c-incompressible.*

Proof. 1. Let c' be such that $K(x) \leq |x| + c'$. Observe that there exists c'' such that $K(0^n u) \leq K(u) + c''$ hence

$$K(0^n u) \leq n + c' + c'' \leq \frac{1}{2}|0^n u| + c' + c''$$

So that $K(0^n u) \geq |0^n u| - c$ is impossible for n large enough.

2. Same argument: There exists c'' such that, for any palindrome x,

$$K(x) \leq \frac{1}{2}|x| + c''$$

3. The proof follows the classical argument to get the law of large numbers (cf. Feller's book [15]). Let us do it for $\alpha = \frac{2}{3}$, so that $\frac{\alpha}{2} = \frac{1}{3}$.

Let A_n be the set of strings of length n with $\leq \frac{n}{3}$ zeros. We estimate the number N of elements of A_n.

$$N = \sum_{i=0}^{i=\frac{n}{3}} \binom{n}{i} \leq (\frac{n}{3} + 1) \binom{n}{\frac{n}{3}} = (\frac{n}{3} + 1) \frac{n!}{\frac{n}{3}!\ \frac{2n}{3}!}$$

Use inequality $1 \leq e^{\frac{1}{12n}} \leq 1.1$ and Stirling's formula (1730),

$$\sqrt{2n\pi}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n+1}} < n! < \sqrt{2n\pi}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n}}$$

Observe that $1.1\ (\frac{n}{3}+1) < n$ for $n \geq 2$. Therefore,

$$N < n\frac{\sqrt{2n\pi}\left(\frac{n}{e}\right)^n}{\sqrt{2\frac{n}{3}\pi}\left(\frac{\frac{n}{3}}{e}\right)^{\frac{n}{3}}\sqrt{2\frac{2n}{3}\pi}\left(\frac{\frac{2n}{3}}{e}\right)^{\frac{2n}{3}}} = \frac{3}{2}\sqrt{\frac{n}{\pi}}\left(\frac{3}{\sqrt[3]{4}}\right)^n$$

Using Proposition 1.16, for any element of A_n, we have

$$K(x \mid n) \leq \log(N) + d \leq n\log\left(\frac{3}{\sqrt[3]{4}}\right) + \frac{\log n}{2} + d$$

Since $\frac{27}{4} < 8$, we have $\frac{3}{\sqrt[3]{4}} < 2$ and $\log\left(\frac{3}{\sqrt[3]{4}}\right) < 1$. Hence, $n - c \leq n\log\left(\frac{3}{\sqrt[3]{4}}\right) + \frac{\log n}{2} + d$ is impossible for n large enough.
So that x cannot be c-incompressible. □

Let us give a common framework to the three above examples so as to get some flavor of what can be a statistical test. To do this, we follow the above proofs of compressibility.

Example 4.7.

1. [Constant left half length prefix]
Set $V_m =$ all strings with m zeros ahead. The sequence $V_0, V_1, \ldots$ is decreasing. The number of strings of length n in V_m is 0 if $m > n$ and 2^{n-m} if $m \leq n$. Thus, the proportion $\frac{\sharp\{x \mid |x|=n\ \wedge\ x \in V_m\}}{2^n}$ of length n words which are in V_m is 2^{-m}.

2. [Palindromes] *Put in V_m all strings which have equal length m prefix and suffix. The sequence $V_0, V_1, \ldots$ is decreasing. The number of strings of length n in V_m is 0 if $m > \frac{n}{2}$ and 2^{n-2m} if $m \leq \frac{n}{2}$. Thus, the proportion of length n words which are in V_m is 2^{-2m}.*

3. [0 and 1 not equidistributed] *Put in $V_m^\alpha =$ all strings x such that the number of zeros is $\leq (\alpha + (1-\alpha)2^{-m})\frac{|x|}{2}$. The sequence $V_0, V_1, \ldots$ is decreasing. A computation analogous to that done in the proof of the law of large numbers shows that the proportion of length n words which are in V_m is $\leq 2^{-\gamma m}$ for some $\gamma > 0$ (independent of m).*

Now, what about other statistical tests? But what is a statistical test? A convincing formalization has been developed by Martin-Löf. The intuition is that illustrated in Example 4.7 augmented of the following feature: each V_m is computably enumerable and so is the relation $\{(m,x) \mid x \in V_m\}$. A feature which is analogous to the partial computability assumption in the definition of Kolmogorov complexity.

Definition 4.8. *[Abstract notion of statistical test, Martin-Löf, 1964] A statistical test is a family of nested critical sets*

$$\{0,1\}^* \supseteq V_0 \supseteq V_1 \supseteq V_2 \supseteq ... \supseteq V_m \supseteq ...$$

such that $\{(m,x) \mid x \in V_m\}$ *is computably enumerable and the proportion* $\frac{\sharp\{x \mid |x|=n \ \wedge \ x \in V_m\}}{2^n}$ *of length* n *words which are in* V_m *is* $\leq 2^{-m}$.

Intuition. The bound 2^{-m} *is just a normalization. Any bound* $b(n)$ *such that* $b : \mathbb{N} \to \mathbb{Q}$ *which is computable, decreasing and with limit* 0 *could replace* 2^{-m}. *The significance of* $x \in V_m$ *is that the hypothesis* x is random *is rejected with significance level* 2^{-m}.

Remark. 4.9. Instead of sets V_m one can consider a function $\delta : \{0,1\}^* \to \mathbb{N}$ such that $\frac{\sharp\{x \mid |x|=n \ \wedge \ \delta(x) \geq m\}}{2^n} \leq 2^{-m}$ and δ is computable from below, i.e., $\{(m,x) \mid \delta(x) \geq m\}$ is recursively enumerable.

We have just argued on some examples that all statistical tests from practice are of the form stated by Definition 4.8. Now comes Martin-Löf fundamental result about statistical tests which is in the vein of the invariance theorem.

Theorem 4.10 (Martin-Löf, 1965)**.** *Up to a constant shift, there exists a largest statistical test* $(U_m)_{m \in \mathbb{N}}$

$$\forall (V_m)_{m \in \mathbb{N}} \quad \exists c \quad \forall m \quad V_{m+c} \subseteq U_m$$

In terms of functions, up to an additive constant, there exists a largest statistical test Δ

$$\forall \delta \quad \exists c \quad \forall x \quad \delta(x) < \Delta(x) + c$$

Proof. Consider $\Delta(x) = |x| - K(x \mid |x|) - 1$.
$\boxed{\Delta \textit{ is a test.}}$ Clearly, $\{(m,x) \mid \Delta(x) \geq m\}$ is computably enumerable.
$\Delta(x) \geq m$ means $K(x \mid |x|) \leq |x| - m - 1$. So no more elements in $\{x \mid \Delta(x) \geq m \ \wedge \ |x| = n\}$ than programs of length $\leq n - m - 1$, which is $2^{n-m} - 1$.
$\boxed{\Delta \textit{ is largest.}}$ x is determined by its rank in the set $V_{\delta(x)} = \{z \mid \delta(z) \geq \delta(x) \ \wedge \ |z| = |x|\}$. Since this set has $\leq 2^{n-\delta(x)}$ elements, the rank of x has a binary representation of length $\leq |x| - \delta(x)$. Add useless zeros ahead to get a word p with length $|x| - \delta(x)$.
With p we get $|x| - \delta(x)$. With $|x| - \delta(x)$ and $|x|$ we get $\delta(x)$ and construct $V_{\delta(x)}$. With p we get the rank of x in this set, hence we get x. Thus,
$K(x \mid |x|) \leq |x| - \delta(x) + c$, i.e., $\delta(x) < \Delta(x) + c$. □

The importance of the previous result is the following corollary which insures that, for words, incompressibility implies (hence is equivalent to) randomness.

Corollary 4.11 (Martin-Löf, 1965)**.** *Incompressibility passes all statistical tests. I.e., for all c, for all statistical test $(V_m)_m$, there exists d such that*

$$\forall x\ (x \textit{ is } c\textit{-incompressible} \ \Rightarrow\ x \notin V_{c+d})$$

> *Proof.* Let x be length conditional c-incompressible. This means that $K(x \mid |x|) \geq |x| - c$. Hence $\Delta(x) = |x| - K(x \mid |x|) - 1 \leq c - 1$, which means that $x \notin U_c$.
> Let now $(V_m)_m$ be a statistical test. Then there is some d such that $V_{m+d} \subseteq U_m$ Therefore $x \notin V_{c+d}$. □

Remark. 4.12. Observe that incompressibility is a *bottom-up* notion: we look at the value of $K(x)$ (or that of $K(x \mid |x|)$).
On the opposite, passing statistical tests is a *top-down* notion. To pass all statistical tests amounts to an inclusion in an intersection: namely, an inclusion in

$$\bigcap_{(V_m)_m} \bigcup_c V_{m+c}$$

4.6 Shortest programs are random finite strings

Observe that optimal programs to compute any object are examples of random strings. More precisely, the following result holds.

Proposition 4.13. *Let $\mathcal{O}$ be an elementary set (cf. Definition 1.9) and $U : \{0,1\}^* \to \{0,1\}^*$, $V : \{0,1\}^* \to \mathcal{O}$ be some fixed optimal functions. There exists a constant c such that, for all $a \in \mathcal{O}$, for all $p \in \{0,1\}^*$, if $V(p) = a$ and $K_V(a) = |p|$ then $K_U(p) \geq |p| - c$. In other words, for any $a \in \mathcal{O}$, if p is a shortest program which outputs a then p is c-random.*

Proof. Consider the function $V \circ U : \{0,1\}^* \to \mathcal{O}$. Using the invariance theorem, let c be such that $K_V \leq K_{V \circ U} + c$. Then, for every $q \in \{0,1\}^*$,

$$\begin{aligned} U(q) = p \quad &\Rightarrow \quad V \circ U(q) = a \\ &\Rightarrow \quad |q| \geq K_{V \circ U}(a) \geq K_V(a) - c = |p| - c \end{aligned}$$

Which proves that $K_U(p) \geq |p| - c$. □

4.7 Random finite strings and lower bounds for computational complexity

Random finite strings (or rather c-incompressible strings) have been extensively used to prove lower bounds for computational complexity, cf. the pioneering paper [42] by Wolfgang Paul, 1979, (see also an account of the proof in our survey paper [16]) and the work by Li & Vitanyi, [31]. The key idea is that a random string can be used as a worst possible input.

5 Formalization of randomness: infinite objects

We shall stick to infinite sequences of zeros and ones: $\{0,1\}^{\mathbb{N}}$.

5.1 Martin-Löf top-down approach with topology and computability

5.1.1 The naive idea badly fails

The naive idea of a random element of $\{0,1\}^{\mathbb{N}}$ is that of a sequence α which is in no set of measure 0. Alas, α is always in the singleton set $\{\alpha\}$ which has measure 0 !

5.1.2 Martin-Löf's solution: effectivize

Martin-Löf's solution to the above problem is to effectivize, i.e., to consider the sole effective measure zero sets.
This approach is, in fact, an extension to infinite sequences of the one Martin-Löf developed for finite objects, cf. §4.5.

Let us develop a series of observations which leads to Martin-Löf's precise solution, i.e., what does mean effective for measure 0 sets.
To prove a probability law amounts to prove that a certain set X of sequences has probability one. To do this, one has to prove that the complement set $Y = \{0,1\}^{\mathbb{N}} \setminus X$ has probability zero. Now, in order to prove that $Y \subseteq \{0,1\}^{\mathbb{N}}$ has probability zero, basic measure theory tells us that one has to include Y in open sets with arbitrarily small probability. I.e., for each $n \in \mathbb{N}$ one must find an open set $U_n \supseteq Y$ which has probability $\leq \frac{1}{2^n}$.
If things were on the real line $\mathbb{R}$ we would say that U_n is a countable union of intervals with rational endpoints.
Here, in $\{0,1\}^{\mathbb{N}}$, U_n is a countable union of sets of the form $u\{0,1\}^{\mathbb{N}}$ where u is a finite binary string and $u\{0,1\}^{\mathbb{N}}$ is the set of infinite sequences which extend u.
In order to prove that Y has probability zero, for each $n \in \mathbb{N}$ one must find a family $(u_{n,m})_{m\in\mathbb{N}}$ such that $Y \subseteq \bigcup_m u_{n,m}\{0,1\}^{\mathbb{N}}$ and $Proba(\bigcup_m u_{n,m}\{0,1\}^{\mathbb{N}}) \leq \frac{1}{2^n}$ for each $n \in \mathbb{N}$.
Now, Martin-Löf makes a crucial observation: mathematical probability laws which we consider necessarily have some effective character. And this effectiveness should reflect in the proof as follows: *the doubly indexed sequence $(u_{n,m})_{n,m\in\mathbb{N}}$ is computable.*

Thus, the set $\bigcup_m u_{n,m}\{0,1\}^{\mathbb{N}}$ is a *computably enumerable open set* and $\bigcap_n \bigcup_m u_{n,m}\{0,1\}^{\mathbb{N}}$ is a countable intersection of a *computably enumerable family of open sets.*

Now comes the essential theorem, which is completely analogous to Theorem 4.10.

Definition 5.1 (Martin-Löf, [32], 1966). *A constructively null G_δ set is any set of the form*

$$\bigcap_n \bigcup_m u_{n,m}\{0,1\}^{\mathbb{N}}$$

where Proba$(\bigcup_m u_{n,m}\{0,1\}^{\mathbb{N}}) \leq \frac{1}{2^n}$ *(which implies that the intersection set has probability zero) and the sequence* $u_{n,m}$ *is computably enumerable.*

Theorem 5.2 (Martin-Löf, [32], 1966). *There exist a largest constructively null G_δ set*

Let us insist that the theorem says *largest*, up to nothing, really largest relative to set inclusion.

Definition 5.3 (Martin-Löf, [32], 1966). *A sequence $\alpha \in \{0,1\}^{\mathbb{N}}$ is Martin-Löf random if it belongs to no constructively null G_δ set (i.e., if it does not belongs to the largest one).*

In particular, the family of random sequences, being the complement of a constructively null G_δ set, has probability 1. And the observation above Definition 5.1 insures that Martin-Löf random sequences satisfy all usual probabilities laws. Notice that *the last statement can be seen as an improvement of all usual probabilities laws: not only such laws are true with probability* 1 *but they are true for all sequences in the measure* 1 *set of Martin-Löf random sequences.*

5.2 The bottom-up approach

5.2.1 The naive idea badly fails

Another natural naive idea to get randomness for sequences is to extend randomness from finite objects to infinite ones. The obvious proposal is to consider sequences $\alpha \in \{0,1\}^{\mathbb{N}}$ such that, for some c,

$$\forall n \quad K(\alpha \restriction n) \geq n - c \tag{1}$$

However, Martin-Löf proved that there is no such sequence.

Theorem 5.4 (Large oscillations (Martin-Löf, [33], 1971)). *If $f : \mathbb{N} \to \mathbb{N}$ is computable and $\sum_{n \in \mathbb{N}} 2^{-f(n)} = +\infty$ then, for every $\alpha \in \{0,1\}^{\mathbb{N}}$, there are infinitely many k such that $K(\alpha \restriction k) \leq k - f(k) - O(1)$.*

> *Proof.* Let us do the proof in the case $f(n) = \log n$ which is quite limpid (recall that the harmonic series $\frac{1}{n} = 2^{-\log n}$ has infinite sum). Let k be any integer. The word $\alpha \restriction k$ prefixed with 1 is the binary representation of an integer n (we put 1 ahead of $\alpha \restriction k$ in order to avoid a first block of non significative zeros). We claim that $\alpha \restriction n$ can be recovered from $\alpha \restriction [k+1, n]$ only. In fact,
>
> - $n - k$ is the length of $\alpha \restriction [k+1, n]$,

- $k = \lfloor \log n \rfloor + 1 = \lfloor \log(n-k) \rfloor + 1 + \varepsilon$ (where $\varepsilon \in \{0,1\}$) is known from $n-k$ and ε,
- $n = (n-k) + k$.
- $\alpha \restriction k$ is the binary representation of n.

The above analysis describes a computable map $f : \{0,1\}^* \times \{0,1\} \to \{0,1\}^*$ such that $\alpha \restriction n = f(\alpha \restriction [k+1, n], \varepsilon)$. Applying Proposition 1.15, point 3, we get

$$K(\alpha \restriction n) \leq K(\alpha \restriction [k+1, n]) + O(1) \leq n - k + O(1) = n - \log(n) + O(1)$$

□

5.2.2 Miller & Yu's theorem

It took about forty years to get a characterization of randomness via Kolmogorov complexity which completes Theorem 5.4 in a very pleasant and natural way.

Theorem 5.5 (Miller & Yu, [34], 2008)**.** *The following conditions are equivalent:*

i. *The sequence $\alpha \in \{0,1\}^{\mathbb{N}}$ is Martin-Löf random*

ii. $\exists c \quad \forall k \quad K(\alpha \restriction k) \geq k - f(k) - c$ *for every total computable function $f : \mathbb{N} \to \mathbb{N}$ satisfying $\sum_{n \in \mathbb{N}} 2^{-f(n)} < +\infty$*

iii. $\exists c \quad \forall k \quad K(\alpha \restriction k) \geq k - H(k) - c$

Moreover, there exists a particular total computable function $g : \mathbb{N} \to \mathbb{N}$ satisfying $\sum_{n \in \mathbb{N}} 2^{-g(n)} < +\infty$ such that one can add a fourth equivalent condition:

iv. $\exists c \quad \forall k \quad K(\alpha \restriction k) \geq k - g(k) - c$

Recently, an elementary proof of this theorem was given by Bienvenu, Merkle & Shen in [3], 2008. Equivalence $i \Leftrightarrow iii$ is due to Gács, [19], 1980.

5.2.3 Variants of Kolmogorov complexity and randomness

Bottom-up characterization of random sequences have been obtained using Levin monotone complexity, Schnorr process complexity and prefix complexity (cf. §3.1, §3.2 and §3.3).

Theorem 5.6. *The following conditions are equivalent:*

i. *The sequence $\alpha \in \{0,1\}^{\mathbb{N}}$ is Martin-Löf random*

ii. $\exists c \quad \forall k \quad |K^{mon}(\alpha \restriction k) - k| \leq c$

iii. $\exists c \quad \forall k \quad |S(\alpha \restriction k) - k| \leq c$

iv. $\exists c \quad \forall k \quad H(\alpha \restriction k) \geq k - c$

Equivalence $i \Leftrightarrow ii$ is due to Levin ([52], 1970). Equivalence $i \Leftrightarrow iii$ is due to Schnorr ([45], 1971). Equivalence $i \Leftrightarrow iv$ is due to Schnorr and Chaitin ([10], 1975).

5.3 Randomness: a robust mathematical notion

Besides the top-down definition of Martin-Löf randomness, we mentioned above diverse bottom-up characterizations via properties of the initial segments with respect to variants of Kolmogorov complexity. There are other top-down and bottom-up characterizations, we mention two of them in this §.
This variety of characterizations shows that Martin-Löf randomness is a robust mathematical notion.

5.3.1 Randomness and martingales

Recall that a martingale is a function $d : \{0,1\}^* \to \mathbb{R}^+$ such that

$$\forall u \quad d(u) = \frac{d(u0) + d(u1)}{2}$$

The intuition is that a player tries to predict the bits of a sequence $\alpha \in \{0,1\}^{\mathbb{N}}$ and bets some amount of money on the values of these bits. If his guess is correct he doubles his stake, else he looses it. Starting with a positive capital $d(\varepsilon)$ (where ε is the empty word), $d(\alpha \upharpoonright k)$ is his capital after the k first bits of α have been revealed.
The martingale d wins on $\alpha \in \{0,1\}^{\mathbb{N}}$ if the capital of the player tends to $+\infty$.
The martingale d is computably approximable from below if the left cut of $d(u)$ is computably enumerable, uniformly in u (i.e., $\{(u,q) \in \{0,1\}^* \times \mathbb{Q} \mid q \leq d(u)\}$ is c.e.).

Theorem 5.7 (Schnorr, [46], 1971)**.** *A sequence $\alpha \in \{0,1\}^{\mathbb{N}}$ is Martin-Löf random if and only if no martingale computably approximable from below wins on α.*

5.3.2 Randomness and compressors

Recently, Bienvenu & Merkle obtained quite remarkable characterizations of random sequences in the vein of Theorems 5.6 and 5.5 involving *computable* upper bounds of K and H.

Definition 5.8. *A compressor is any partial computable $\Gamma : \{0,1\}^* \to \{0,1\}^*$ which is one-to-one and has computable domain. A compressor is said to be prefix-free if its range is prefix-free.*

Proposition 5.9.

1. If Γ is a compressor (resp. a prefix-free compressor) then

$$\exists c \quad \forall x \in \{0,1\}^* \quad K(x) \leq |\Gamma(x)| + c \qquad \textit{(resp. } H(x) \leq |\Gamma(x)| + c\textit{)}$$

2. For any computable upper bound F of K (resp. of H) there exists a compressor (resp. a prefix-free compressor) Γ such that

$$\exists c \quad \forall x \in \{0,1\}^* \quad |\Gamma(x)| \leq F(x) + c$$

Now comes the surprising characterizations of randomness in terms of *computable functions.*

Theorem 5.10 (Bienvenu & Merkle, [2], 2007)**.** *The following conditions are equivalent:*

i. *The sequence* $\alpha \in \{0,1\}^{\mathbb{N}}$ *is Martin-Löf random*

ii. *For all prefix-free compressor* $\Gamma : \{0,1\}^* \to \{0,1\}^*$,

$$\exists c \quad \forall k \quad |\Gamma(\alpha \restriction k)| \geq k - c$$

iii. *For all compressor* Γ, $\exists c \quad \forall k \quad |\Gamma(\alpha \restriction k)| \geq k - H(k) - c$

Moreover, there exists a particular prefix-free compressor Γ^* *and a particular compressor* $\Gamma^{\#}$ *such that one can add two more equivalent conditions:*

iv. $\exists c \quad \forall k \quad |\Gamma^*(\alpha \restriction k)| \geq k - c$

v. $\exists c \quad \forall k \quad |\Gamma^{\#}(\alpha \restriction k)| \geq k - |\Gamma^*(\alpha \restriction k)| - c$

5.4 Randomness: a fragile property

Though the notion of Martin-Löf randomness is robust, with a lot of equivalent definitions, as a property, it is quite fragile.
In fact, random sequences loose their random character under very simple computable transformation. For instance, even if $a_0a_1a_2...$ is random, the sequence $0a_00a_10a_20...$ IS NOT random since it fails the following Martin-Löf test:

$$\bigcap_{n\in\mathbb{N}} \{\alpha \mid \forall i < n \ \alpha(2i+1) = 0\}$$

Indeed, $\{\alpha \mid \forall i < n \ \alpha(2i+1) = 0\}$ has probability 2^{-n} and is an open subset of $\{0,1\}^{\mathbb{N}}$.

5.5 Randomness is not chaos

In a series of papers [37, 38, 39], 1993-1996, Joan Rand Moschovakis introduced a very convincing notion of chaotic sequence $\alpha \in \{0,1\}^{\mathbb{N}}$. It turns out that the set of such sequences has measure zero and is disjoint from Martin-Löf random sequences.
This stresses that *randomness is not chaos.* As mentioned in §5.1.2, random sequences obey laws, those of probability theory.

5.6 Oracular randomness

5.6.1 Relativization

Replacing "computable" by "computable in some oracle", all the above theory relativizes in an obvious way, using oracular Kolmogorov complexity and the

oracular variants.
In particular, when the oracle is the halting problem, i.e. the computably enumerable set $\emptyset'$, the obtained randomness is called 2-randomness.
When the oracle is the halting problem of partial $\emptyset'$-computable functions, i.e. the computably enumerable set $\emptyset''$, the obtained randomness is called 3-randomness. And so on.
Of course, 2-randomness implies randomness (which is also called 1-randomness) and 3-randomness implies 2-randomness. And so on.

5.6.2 Kolmogorov randomness and $\emptyset'$

A natural question following Theorem 5.4 is to look at the so-called *Kolmogorov random sequences* which satisfy $K(\alpha \upharpoonright k) \geq k - O(1)$ for infinitely many k's. This question got a very surprising answer involving 2-randomness.

Theorem 5.11 (Nies, Stephan & Terwijn, [41], 2005)**.** *Let $\alpha \in \{0,1\}^{\mathbb{N}}$. There are infinitely many k such that, for a fixed c, $K(\alpha \upharpoonright k) \geq k - c$ (i.e., α is Kolmogorov random) if and only if α is 2-random.*

5.7 Randomness: a new foundation for probability theory?

Now that there is a sound mathematical notion of randomness, is it possible/reasonable to use it as a new foundation for probability theory?
Kolmogorov has been ambiguous on this question. In his first paper on the subject, see p. 35–36 of [25], 1965, he briefly evoked that possibility :

> *...to consider the use of the [Algorithmic Information Theory] constructions in providing a new basis for Probability Theory.*

However, later, see p. 35–36 of [26], 1983, he separated both topics:

> *"there is no need whatsoever to change the established construction of the mathematical probability theory on the basis of the general theory of measure. I am not enclined to attribute the significance of necessary foundations of probability theory to the investigations [about Kolmogorov complexity] that I am now going to survey. But they are most interesting in themselves.*

though stressing the role of his new theory of random objects for *mathematics as a whole* in [26], p. 39:

> *The concepts of information theory as applied to infinite sequences give rise to very interesting investigations, which, without being indispensable as a basis of probability theory, can acquire a certain value in the investigation of the algorithmic side of mathematics as a whole.*

References

[1] Becher V., Figueira S., Nies A., Picchi S. and Vitányi P. Program size complexity for possibly infinite computations. *Notre Dame Journal of Formal Logic*, 46(1):51–64, 2005.

[2] Bienvenu L. & Merkle W. Reconciling data compression and Kolmogorov complexity. *ICALP 2007, LNCS 4596*, 643–654, 2007.

[3] Bienvenu L., Merkle W. & Shen A. A simple proof of Miller-Yu theorem. *Fundamenta Informaticae*, 83(1-2):21–24, 2008.

[4] Bonfante G., Kaczmarek M. & Marion J-Y. On abstract computer virology: from a recursion-theoretic perspective. *Journal of computer virology*, 3-4, 2006.

[5] Calude C. & Jürgensen H. Is complexity a source of incompleteness? *Advances in Applied Mathematics*, 35:1-15, 2005.

[6] Chaitin G. On the length of programs for computing finite binary sequences. *Journal of the ACM*, 13:547–569, 1966.

[7] Chaitin G. On the length of programs for computing finite binary sequences: statistical considerations. *Journal of the ACM*, 16:145–159, 1969.

[8] Chaitin G. Computational complexity and gödel incompleteness theorem. *ACM SIGACT News*, 9:11–12, 1971.

[9] Chaitin G. Information theoretic limitations of formal systems. *Journal of the ACM*, 21:403–424, 1974.

[10] Chaitin G. A theory of program size formally identical to information theory. *Journal of the ACM*, 22:329–340, 1975.

[11] J.P. Delahaye. *Information, complexité, hasard*. Hermès, 1999 (2d edition).

[12] J.P. Delahaye. *Complexités : Aux limites des mathématiques et de l'informatique*. Belin-Pour la Science, 2006.

[13] Durand B. & Zvonkin A. Complexité de Kolmogorov. *L'héritage de Kolmogorov en mathématiques*. E. Charpentier, A. Lesne, N. Nikolski (eds). Belin, p. 269–287, 2004.

[14] Fallis D. The source of Chaitin's incorrectness. *Philosophia Mathematica*, 4: 261-269, 1996.

[15] Feller W. *Introduction to probability theory and its applications*, volume 1. John Wiley, 1968 (3d edition).

[16] Ferbus-Zanda M. & Grigorieff S. Is randomnes native to computer science? *Current Trends in Theoretical Computer Science.* G. Paun, G. Rozenberg, A. Salomaa (eds.). World Scientific, pages 141–179, 2004.

[17] Ferbus-Zanda M. & Grigorieff S. Kolmogorov complexity and set theoretical representations of integers. *Math. Logic Quarterly*, 52(4):381–409, 2006.

[18] Gács P. On the symmetry of algorithmic information. *Soviet Math. Dokl.*, 15:1477–1480, 1974.

[19] Gács P. Exact expressions for some randomness tests. *Zeitschrift für Math. Logik u. Grundlagen der Math.*, 26:385–394, 1980.

[20] Gács P. Lectures notes on descriptional complexity and randomness. *Boston University*, pages 1–67, 1993.
http://cs-pub.bu.edu/faculty/gacs/Home.html.

[21] Huffman D.A. A method for construction of minimum-redundancy codes. *Proceedings IRE*, 40:1098–1101, 1952.

[22] Knuth D. *The Art of Computer Programming. Volume 2: semi-numerical algorithms.* Addison-Wesley, 1981 (2d edition).

[23] Kolmogorov A.N. *Grundbegriffe der Wahscheinlichkeitsrechnung.* Springer-Verlag, 1933. English translation: *Foundations of the Theory of Probability*, Chelsea, 1956.

[24] Kolmogorov A.N. On tables of random numbers. *Sankhya, The Indian Journal of Statistics, ser. A*, 25:369–376, 1963.

[25] Kolmogorov A.N. Three approaches to the quantitative definition of information. *Problems Inform. Transmission*, 1(1):1–7, 1965.

[26] Kolmogorov A.N. Combinatorial foundation of information theory and the calculus of probability. *Russian Math. Surveys*, 38(4):29–40, 1983.

[27] Lacombe D. La théorie des fonctions récursives et ses applications. *Bull. Société Math. de France*, 88:393–468, 1960.

[28] van Lambalgen M. Algorithmic information theory. *The Journal of Symbolic Logic*, 54(4):1389-1400, 1989.

[29] Levin L. On the notion of a random sequence. *Soviet Math. Dokl.*, 14:1413–1416, 1973.

[30] Levin L. Laws of information conservation (non-growth) and aspects of the foundation of probability theory. *Problems Inform. Transmission*, 10(3):206–210, 1974.

[31] Li M. & Vitányi P. *An introduction to Kolmogorov Complexity and its applications.* Springer, 2d Edition, 1997.

[32] Martin-Löf P. The definition of random sequences. *Information and Control*, 9:602–619, 1966.

[33] Martin-Löf P. Complexity of oscilations in infinite binary sequences. *Z. Wahrscheinlichkeitstheorie verw. Geb.*, 19:225–230, 1971.

[34] Miller J.& Yu L. On initial segment complexity and degrees of randomness. *Trans. Amer. Math. Soc.* , to appear.

[35] von Mises R. Grundlagen der wahrscheinlichkeitsrechnung. *Mathemat. Zeitsch.*, 5:52–99, 1919.

[36] von Mises R. *Probability, Statistics and Truth.* Macmillan, 1939. Reprinted: Dover, 1981.

[37] Moschovakis J.R. An intuitionistic theory of lawlike, choice and lawless sequences. *Logic Colloquium '90.* J. Oikkonen and J. Väänänen (eds.). Lecture Notes in Logic 2 (Springer, Berlin), 191–209, 1993.

[38] Moschovakis J.R. More about relatively lawless sequences. *The Journal of Symbolic Logic*, 59(3):813–829, 1994.

[39] Moschovakis J.R. A classical view of the intuitionistic continuum. *Annals of Pure and Applied Logic*, 81:9–24, 1996.

[40] von Neumann J. Various techniques used in connection with random digits. *Monte Carlo Method*, Householder A.S., Forsythe G.E. & Germond H.H., eds., National Bureau of Standards Applied Mathematics Series (Washington, D.C.: U.S. Government Printing Office), 12:36–38, 1951.

[41] Nies A., Stephan F.& Terwijn S.A. Randomness, relativization and Turing degrees. To appear.

[42] Paul W. Kolmogorov's complexity and lower bounds. *In Proc. 2nd Int. Conf. Fundamentals of Computation Theory*, L. Budach ed., Akademie Verlag, 325–334, 1979.

[43] Raatikainen P. On interpreting Chaitin's Incompleteness theorem. *Journal of Philosophical Logic*, 27(6):569-586, 1998.

[44] Russell B. Mathematical logic as based on the theory of types. *Amer. J. Math.*, 30:222–262, 1908. Reprinted in 'From Frege to Gödel A source book in mathematical logic, 1879-1931', J. van Heijenoort ed., p. 150-182, 1967.

[45] Schnorr P. A unified approach to the definition of random sequences. *Math. Systems Theory*, 5:246–258, 1971.

[46] Schnorr P. Zufälligkeit und Wahrscheinlichkeit. *Lecture Notes in Mathematics, vol. 218*, 1971.

[47] Schnorr P. A Process complexity and effective random tests. *J.of Computer and System Sc.*, 7:376–388, 1973.

[48] Shannon C.E. The mathematical theory of communication. *Bell System Tech. J.*, 27:379–423, 1948.

[49] Soare R. Computability and Recursion. *Bulletin of Symbolic Logic*, 2:284–321, 1996.

[50] Solomonoff R. A formal theory of inductive inference, part I. *Information and control*, 7:1–22, 1964.

[51] Solomonoff R. A formal theory of inductive inference, part II. *Information and control*, 7:224–254, 1964.

[52] Zvonkin A. & Levin L. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Math. Surveys*, 6:83–124, 1970.