Article

An Algorithmic Approach to Emergence

Charles Alexandre Bédard [1,*,†] and Geoffroy Bergeron [2,†]

1   Faculty of Informatics, Università della Svizzera Italiana, 6900 Lugano, Switzerland
2   Centre de Recherches Mathématiques, Université de Montréal, Montréal, QC H3T 1J4, Canada; geoffroy.bergeron@umontreal.ca
*   Correspondence: charles.alexandre.bedard@usi.ch
†   These authors contributed equally to this work.

Abstract: We suggest a quantitative and objective notion of *emergence.* Our proposal uses algorithmic information theory as a basis for an objective framework in which a bit string encodes observational data. A plurality of drops in the Kolmogorov structure function of such a string is seen as the hallmark of emergence. Our definition offers some theoretical results, in addition to extending the notions of coarse-graining and boundary conditions. Finally, we confront our proposal with applications to dynamical systems and thermodynamics.

1. Introduction

Emergence is a concept often referred to in the study of complex systems. Coined in 1875 by the philosopher George H. Lewes in his book *Problems of Life and Mind* [1], the term has ever since mainly been used in qualitative discussions [2,3]. In most contexts, emergence refers to the phenomenon by which novel properties arise in a complex system which is composed of a large quantity of simpler subsystems that do not exhibit those novel properties by themselves, but rather through their collective interactions. The following citation from Wikipedia [4] reflects this popular idea: "For instance, the phenomenon of life as studied in biology is an emergent property of chemistry, and psychological phenomena emerge from the neurobiological phenomena of living things".

For claims such as the above to have a precise meaning, an objective definition of emergence must be provided. Current definitions are framed around a qualitative evaluation of the "novelty" of properties exhibited by a system with respect to those of its constituent subsystems. This state of matters renders generic use of the term ambiguous and subjective, and hence potentially problematic within a scientific discussion. In this paper, we attempt to free the notion of emergence from subjectivity by proposing a mathematical and objective notion of emergence.

1.1. Existing Notions of Emergence

We review a few of the many appeals to the notion of emergence. One of them goes all the way back to Aristotle's metaphysics [5]: "The whole is something over and above its parts, and not just the sum of them all...". This common idea is revisited by the theoretical physicist Philip W. Anderson [6], who claims that "[...] the whole becomes not only more, but very different from the sum of its parts". In the same essay, he highlights the asymmetry between reducing and constructing:

> The ability to reduce everything to simple fundamental laws does not imply the ability to start from those laws and reconstruct the universe. In fact, the more elementary particle physicists tell us about the nature of the fundamental laws, the less relevance they seem to have for the very real problems of the rest of science, much less to those of society.

> The constructionist hypothesis breaks down when confronted with the twin difficulty of scale and complexity. [. . .] at each level of complexity, entirely new properties appear, and the understanding of the new behaviours requires research which I think is as fundamental in its nature as any other. [. . .] At each stage, entirely new laws, concepts, and generalizations are necessary, requiring inspiration and creativity to just as great a degree as the previous one. Psychology is not applied biology, nor biology is applied chemistry.

More recently, David Wallace [7] (Chapter 2) qualifies emergent entities to be "not directly definable in the language of microphysics (try defining a haircut within the Standard Model) but that does not mean that they are somehow independent of that underlying microphysics". Emergent objects are instead seen as patterns, and the existence of those patterns as real things is subjected to a criterion that Wallace attributes to Dennett [8]. The patterns are real if the theories that admit them in their ontology gain in usefulness, in explanatory power or in predictive reliability. For instance, Dennett's criterion mandates that temperature be thought of an emergent but real concept because it is a *useful pattern*. In fact, temperature is not a basic entity of the microphysics, yet a full scientific description of a large system with no reference to the notion temperature completely misses a fundamental aspect. In this spirit, temperature is as real as it is useful, namely, as real as it is used in key phenomenological explanations. Useful patterns, or structures, is a central concept that is formalized within algorithmic information theory, more precisely in non-probabilistic statistics.

1.2. From Systems to Bit Strings

To better appreciate our reliance on algorithmic information theory, we present our epistemological standpoint. We take the realist view that there is a world outside and independent of our perception. This world is made of physical systems, and the goal of science is to understand their properties, their dynamics and their possibilities. This is done through an interplay between the formulation of theories—bold conjectures about how the world is [9]—and their experimental challenges. Pragmatically, theories can serve the purpose of providing simple models to explain the data, an idea which will be explored throughout the paper. Empirical observation, on the other hand, collects data from physical systems. But what is the nature of such data collection? How can one extract from a system, assumed to exist in reality, a string of symbols that can be taken to be binary?

The answer lies in the physics of the measurement process. Observation consists of an interaction between the physical system one cares to learn about and some prepared measurement apparatus. The measurement apparatus then interacts with a computing device (this can be the experimenter) that arranges its memory in a physical representation of a bit string x. A scientist collecting data about a system shall then be left with a string x, which, clearly, is not only determined by the investigated system. The information in x could reflect properties of other systems with which it has previously interacted, such as surrounding systems (the environment), the measurement apparatus and the scientist itself. As observed by Gell-Mann and Lloyd [10], this introduces several sources of arbitrariness into x, in addition to the level of details of the measurement and the coding convention that maps the apparatus's configuration into bits. Furthermore, the knowledge and cognitive biases of the scientist impact *what* is being measured. For Gell-Mann and Lloyd, all this arbitrariness is to be discarded in order to define the (algorithmic) information content of the system through that of x. We do not share this view, as we think that this arbitrariness inhibits a well-posed definition. In fact, a subtlety of scientific investigation concerns how to probe the system in order to push into x the yet-to-be-understood features that it can exhibit; and this shall never be constrained by a fixed method.

Nonetheless, the subjective and error-prone connection between the physical system and the recorded data does not preclude an overall objective modelling of the world. This is because, after all, *the string originates from a real physical process*. Subjectivity and errors impact which process occurs, yet a sufficiently careful analysis of the data finds the best

explanation of its origin, and so also explains the scientist's choices and the unwanted interactions. For instance, if we ask a dishonest scientist to give us data about a system, but he elects instead to give us bits at whim, then investigating the data will lead to models of what was happening in that person's brain, which is itself a part of reality. Thus, the string x is always objective data from a real system, although not necessarily the one that was presumed to be under investigation.

Once the data x is fixed, we face the problem of finding the best explanations for it, which is related to finding its patterns or structures. This is the main investigation of the paper. It can be done in the realm of algorithmic information theory (AIT), a branch of mathematics and logic that offers similar tools as probability theory, but with no need for unexplained randomness. Li and Vitányi, authors of the most cited textbook [11] in the field, claim that "Science may be regarded as the art of data compression" and according to the pioneer Gregory Chaitin [12], "[A] scientific theory is a computer program that enables you to compute or explain your experimental data". Indeed, even theoretical pen and paper work constitutes symbolic manipulations which are inherently algorithmic (see Figure 1).
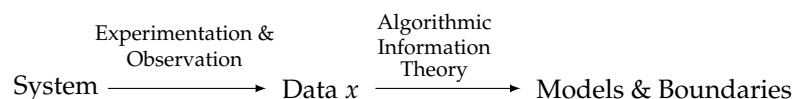

Figure 1. Systems are comprehended through experimentation and observation, which physically yield a bit string. Models and their respective boundaries can then be defined for each string through methods from algorithmic information theory.

A point remains to be addressed. Why work with classical information and computation instead of their quantum counterparts? As quantum computation can be classically emulated [13], the quantum gain is only in speed, and not fundamental in terms of what can or cannot be computed. This work is grounded in computability theory, so by leaving aside questions of time complexity, we also leave aside quantum computation.

1.3. Outline

This paper is organized as follows. In Section 2, we give a review of the basic notions of algorithmic information theory, with a particular focus on non-probabilistic statistics and connections in physics. Building on those, we introduce in Section 3 an algorithmic definition of emergence and we derive from it some concepts and results. In Section 4, we illustrate the relevance of the proposed definition in a toy model as well as in applications to dynamical systems and thermodynamics.

2. A Primer on Algorithmic Methods

Algorithmic information theory (AIT) [14–16] is the mergence of Shannon's theory of information [17] and Turing's theory of computation [18]. Introduced in his seminal paper titled "A Mathematical Theory of Communication", Shannon's theory concerns the ability to communicate a message that comes from a random source of symbols. In this context, the randomness is formalized in the probabilistic setting, and represents ignorance or unpredictability of the symbols to come. The entropy is then a functional on the underlying distribution that quantifies an optimal compression of the message. Concretely, this underlying distribution is often estimated through the observed biases in the frequency of the sequences of symbols to transmit. However, noticing such biases is only a single way to compress a message. For instance, if Alice were to communicate the 10^{10} first digits of π to Bob, a pragmatic application of Shannon's information theory would be of no help since the frequencies of the symbols to transmit are uniform (if π is normal, which it is conjectured to be). However, Alice could simply transmit:

$$\texttt{'The first } 10^{10} \texttt{ digits of } 4\sum_{n=0}^{\infty}\frac{(-1)^n}{2n+1}.\texttt{'}$$

Bob then understands the received message as an instruction that he runs on a universal computing device to obtain the desired message. Equipping information theory with universal computation enables message compression by all possible (computable) means—and not just via statistical biases. As we will see, the length of the best compression of a message is a natural measure of the information contained in the message.

2.1. Algorithmic Complexity

We give the basic definitions and properties of algorithmic complexity. See [11] (Chapters 1–3) for details, attributions and background on computability theory.

The *algorithmic complexity* $K(x)$ of a piece of data x is the length of its shortest computable description. It can be understood as the minimum amount of information required to produce x by any computable process. Per contra to Shannon's notion of information, which supposes an *a priori* random process from which the data has originated, algorithmic complexity is an *intrinsic* measure of information. Because all discrete data can be binary coded, we consider only finite binary strings (referred to as "strings" from now on), i.e.,

$$x \in \{0,1\}^* = \{\epsilon, 0, 1, 00, \dots\},$$

where ϵ stands for the empty word. For a meaningful definition, we have to select a universal computing device $\mathcal{U}$ on which we execute the computation to obtain x from the description. Such a description is called a program p, and since it is itself a string, the length of p is well defined as the number of bits in it and denoted $|p|$. Therefore,

$$K_{\mathcal{U}}(x) := \min_p \{|p| : \mathcal{U}(p) = x\}.$$

Note that abstractly, $\mathcal{U}$ can be thought of as a universal device in any Turing-complete model of computation. In the realm of Turing machines, a universal device expects an input p encoding a pair $p = \langle q, i \rangle$ and simulates the machine of program q on input i. Concretely, $\mathcal{U}$ can be thought of as a modern computer or a human with pen and paper. This is the essence of the Church–Turing thesis, according to which all sufficiently generic approaches to symbolic manipulations are equivalent and encompass physically realizable computations. The invariance theorem for algorithmic complexity guarantees that no other formal mechanism can yield an essentially shorter description. This is because the reference universal computing device $\mathcal{U}$ can simulate any other computing device $\mathcal{V}$ with a constant overhead in program length; i.e., there exists a constant $C_{\mathcal{U}\mathcal{V}}$ such that

$$K_{\mathcal{U}}(x) \le K_{\mathcal{V}}(x) + C_{\mathcal{U}\mathcal{V}}$$

holds uniformly for all x. If $\mathcal{V}$ is also a reference universal computing device, then $K_{\mathcal{U}}(x)$ and $K_{\mathcal{V}}(x)$ differ by at most a fixed constant. It is customary in this field to use the big-O notation and express an error term as a function of $|x| = n$. Henceforth, the character n shall be reserved to denote the length of x. In general, $O(f(n))$ denotes a quantity whose absolute value does not exceed $f(n)$ by more than a fixed multiplicative factor, so in this case, we can write $K_{\mathcal{U}}(x) \le K_{\mathcal{V}}(x) + O(1)$. Throughout the paper, $O(\log n)$ error terms shall often occur. For significant data set, $|x| = n$ is very large and so $O(\log n)$ is comparatively a very small quantity. For instance, if $\alpha \le n$, then $K(\alpha) \le O(\log n)$ since α is a (binary) number of at most $\log n + 1$ bits, and one program to compute α is simply to enumerate all those bits.

Since the ambiguity in the choice of computing devices is lifted (up to an additive constant), we omit the subscript $\mathcal{U}$ in the notation. Algorithmic complexity is in this sense a *universal* measure of the complexity of x.

The *conditional algorithmic complexity* $K(x \mid y)$ of x relative to y is defined as the length of the shortest program to compute x, if y is provided as an auxiliary input. More formally,

$$K(x \mid y) := \min_p \{|p| : \mathcal{U}(p, y) = x\}.$$

Multiple strings $x_1, \ldots, x_n$ can be encoded into a single one denoted $\langle x_1, \ldots, x_n \rangle$. The algorithmic complexity $K(x_1, \ldots, x_n)$ of multiple strings is then defined as

$$K(x_1, \ldots, x_n) := \min_p \{|p| : \mathcal{U}(p) = \langle x_1, \ldots, x_n \rangle\}.$$

For technical reasons, we restrict the set of programs resulting in a halting computation to be such that no halting program is a prefix of another halting program, namely, the set of halting programs is a *prefix code*. One way to impose such a constraint is to have all programs *self-delimiting*, meaning that the computational device $\mathcal{U}$ halts its computation after reading the last bit of the program p, but no further. This restriction is not fundamentally needed for our purposes, but it entails an overall richer and cleaner theory of algorithmic information. For instance, the upcoming relation (1) holds within an additive constant only if self-delimitation is imposed.

A key property of entropy in Shannon's theory is the chain rule that relates the entropy of a pair to those of the constituents. This is also achieved in the realm of AIT. Let x^* be the shortest program that computes x. (If there are more than one "shortest program", then x^* is the fastest, and if more than one have the same running time, then x^* is the first in lexicographic order.) Algorithmic complexity satisfies the important chain rule

$$K(x, y) = K(x) + K(y \mid x^*) + O(1). \tag{1}$$

One obvious procedure to compute the pair of strings x and y is to first compute x out of its shortest program x^*, and then use x^* to compute y, which proves the "$\leq$" part of (1). The "$\geq$" side, harder to prove, states that the previous procedure to compute $\langle x, y \rangle$ is nearly optimal in terms of program length. A looser formulation of the chain rule is

$$K(x, y) = K(x) + K(y \mid x) + O(\log n), \tag{2}$$

where $n = \max(|x|, |y|)$.

A major drawback of algorithmic complexity for pragmatic use is that it is *uncomputable*, namely, no algorithm can return $K(x)$ on a generic input x.

2.2. Non-Probabilistic Statistics

Standard statistics are founded upon probability theory. Remarkably, the same person who axiomatized probability theory managed to detach statistics and model selection from its probabilistic roots. Kolmogorov suggested [19] that AIT could serve as a basis for statistics and model selection for individual data. See [20] for a modern review.

In this setting, a *model* of x is defined to be a finite set $S \subseteq \{0, 1\}^*$ such that $x \in S$. It is also referred to as an *algorithmic statistic* or *non-probabilistic statistic*. Any model S can be quantified by its cardinality, denoted $|S|$, and by its algorithmic complexity $K(S)$, yielding a quantitative meaning of "simple" and "complex". To define $K(S)$ properly, let again $\mathcal{U}$ be the reference universal computing device. Let p be a program that computes an encoding $\langle x_1, \ldots, x_N \rangle$ of the lexicographical ordering of the elements of S and halts.

$$\mathcal{U}(p) = \langle x_1, \ldots, x_N \rangle, \qquad \text{where} \qquad S = \{x_1, \ldots, x_N\}.$$

Then, S^* is the shortest such program and $K(S)$ is its length. When S and S' are two models of x of the same complexity α, we say that S is *a better* model than S' if it contains fewer elements. This is because there is less ambiguity in specifying x within a model containing fewer elements. In this sense, more of the distinguishing properties of x are reflected by such a model. Indeed, among all models of complexity $\leq \alpha$, the model of smallest cardinality is *optimal* for this fixed threshold of complexity.

Any string x of length n exhibits two canonical models shown in Table 1. The first is simply $S_{\text{Babel}} := \{0, 1\}^n$, which has small complexity as it is easy to describe—a program producing it only requires the information about n. However, it is a large set, containing 2^n

elements. It is intuitively a bad model since it does not capture any properties of x, except its length. The other canonical example is $S_x := \{x\}$. This time, S_x has large complexity, namely, it is as hard to describe as x is, but it is a very tiny set with a single element. The model S_x is also bad as it captures *everything* about x, even the noise or incidental randomness. This significantly weighs down the description of the model, and is commonly known as *over-fitting*. A good map of Montreal is not Montreal itself!

Table 1. Complexity and cardinality of S_{Babel} and S_x.

Model S	Complexity	$K(S)$	Cardinality	$\log |S|$
$S_{\text{Babel}} = \{0,1\}^n$	Small	$K(n) + O(1)$	Large	n
$S_x = \{x\}$	Large	$K(x) + O(1)$	Small	0

If S is a model of x, then

$$K(x \mid S) \le \log |S| + O(1),$$

because one way to compute x out of S is to give the $\lceil \log |S| \rceil$ bit-long *index* of x in the lexicographical ordering of the elements of S, where $\lceil \cdot \rceil$ denotes the ceiling function. (Note that the program can be made self-delimiting at no extra cost because the length of the index can be computed from the resource S provided.) This trivial computation of x relative to S is known as the *data-to-model code* (which should really be called model-to-data.). A string x is a *typical* element of its model S if the data-to-model code is essentially the shortest program, i.e., if

$$K(x \mid S) = \log |S| + O(1).$$

In such a case, there is no simple property that singles out x from the other elements of S. Notice also that the data x can always be described by a *two-part description*: the model description and the data-to-model code. Hence,

$$K(x) \le K(S) + \log |S| + O(1). \tag{3}$$

In his seminal paper [21] on the foundations of theoretical (probabilistic) statistics, Fischer stated: "The statistic chosen should summarize the whole of the relevant information supplied by the sample. This may be called the *Criterion of Sufficiency*.". Kolmogorov suggested an algorithmic counterpart. A model $S \ni x$ is *sufficient* for x if the two-part description with S as a model is an almost-shortest description, namely,

$$K(S) + \log |S| \le K(x) + O(\log n).$$

Here, the $O(\log n)$ refers to $K(K(S), \log |S|)$ since the self-delimited two-part code implicitly carries the length of each part as its intrinsic information, while the optimal one-part code, x^*, in general does not know about the size of each part.

Such models constrain the set of strings to those sharing "the whole of the relevant" properties that characterize x, which is then a typical element of those models.

Finally, a good model should not give more than the relevant information supplied by the data. The simplest sufficient model displays all the relevant properties of the data, and nothing more, thus preventing over-fitting. It is called the *minimal sufficient model* and denoted S_M.

Kolmogorov's Structure Function

For a given string x, its associated Kolmogorov's structure function explores trade-offs between the complexity and cardinality of possible models. This function maps any complexity threshold to the log-cardinality of the optimal model S within that threshold. It will be applied to investigate more interesting models, "between" S_{Babel} and S_x.

Definition 1. *(Structure function) The structure function of a string x, $h_x : \mathbb{N} \to \mathbb{N}$, is defined as*

$$h_x(\alpha) = \min_{S \ni x}\{\lceil \log|S| \rceil \ : K(S) \le \alpha\}\,.$$

We say that the optimal model of α bits or less *witness $h_x(\alpha)$*. Extremal points of $h_x(\alpha)$ are essentially determined by S_{Babel} and S_x, as shown in Table 1. It follows that

$$h_x(K(n) + O(1)) \le \log |S_{\text{Babel}}| = n \qquad \text{and} \qquad h_x(K(x) + O(1)) \le \log|S_x| = 0\,.$$

An upper bound for h_x is prescribed by noticing that a more complex model, S', can be built from a previously described one, S, by including into the description of S' the first bits of index of $x \in S$. In this case, for each bit of index specified, the log-cardinality of the resulting model reduces by one. This implies that the overall slope of the structure function must be ≤ -1. (A knowledgeable reader may frown upon this coarse argument because prefix technicalities demand a more careful analysis as is done in Section 3.1. Such an analysis shows that the relations as presented here hold up to logarithmic fluctuations.) Applying this argument to S_{Babel}, we conclude that the graph of $h_x(\alpha)$ is upper-bounded by the line $n + K(n) - \alpha$.

A lower bound is obtained from applying Equation (3) to the model S witnessing $h_x(\alpha)$. In such a case, $K(S) \le \alpha$ and $\log|S| = h_x(\alpha)$, so

$$K(x) - \alpha \le h_x(\alpha)\,.$$

This means that the graph of $h_x(\alpha)$ always sits above the line $K(x) - \alpha$, known as the *sufficiency line*. The above inequality turns into an equality (up to a logarithmic term) if and only if the witness S is a sufficient model, by definition. Thus, this sufficiency line is reached by the structure function when enough bits of model description are available to formulate a sufficient statistics for x. Once the structure function reaches the sufficiency line, it stays near it, within logarithmic precision, because it is then bounded above and below by the -1 slope linear regime. The sufficiency line is always reached because S_x qualifies as a sufficient model.

For concreteness, a plot of $h_x(\alpha)$, for some string x of length n, is given in Figure 2. In this example, the string x is such that optimal models of complexity smaller than α_M are not very insightful; they give as little information about x as a raw recitation of the first bits of x. Indeed, bits describing those models are used as inefficiently as an enumeration of x. In sharp contrast, S_M is exploiting complex structures in x to efficiently constrain the size of the resulting set. It is fundamentally different from the optimal model of $\alpha_M - 1$ bits as it does not recite trivial properties of x, but rather expresses some distinguishing property of the data. Indeed, from α_M bits of model, the uncertainty about x is decreased by much more than α_M bits, as x is now known to belong to a much smaller set. In this example, S_M is the minimal sufficient statistics.

The complexity of the minimal sufficient statistics, α_M, is known as the *sophistication* of the string x, which captures the amount of algorithmic information needed to grasp all structures—or regularities—of the string. Technically, here we refer to set-sophistication, as defined in [22], since sophistication has been originally defined [23] through total functions as model classes instead of finite sets. Importantly, Vitányi has investigated [24] three different classes of model: finite sets, probability distributions (or statistical ensembles, cf. the following section) and total functions. Although they may appear to be of increasing generality, he shows that they are not. Any model of a particular class defines a model in the other two classes of the same complexity (up to a logarithmic term) and log-cardinality (or analogue).
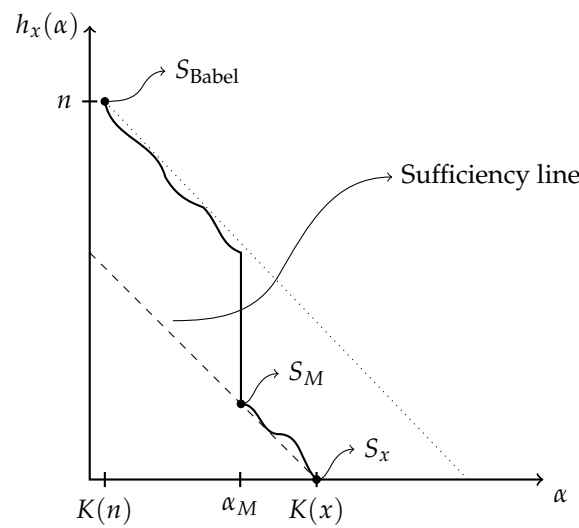
Figure 2. Kolmogorov's structure function of a string x of length $|x| = n$.

Because algorithmic complexity is uncomputable, so is the structure function. However, it can be *upper semi-computed*, which means that there is an algorithm that keeps outputting better upper bounds of the structure function until it eventually reaches the actual structure function. When this happens, the algorithm does not halt, as it keeps looking for better upper bounds, not knowing that this is in vain. In our generic context of finding explanations for observation data, this upper semi-computation can be seen to be realized by the scientific enterprise, which seeks and finds simpler and better models.

2.3. Algorithmic Connections in Physics

Ideas of using AIT and non-probabilistic statistics to enhance the understanding of physical concepts are not new. For example, expanding on the famous Landauer principle, Bennett [25] suggested that thermodynamics is more a theory of computation than a theory of probability, and so better rooted in AIT than in Shannon information theory. Based on his work, Zurek proposed [26] the notion of *physical entropy*, which generalizes thermodynamic entropy to ensure consistency. In the case of a system with microstate x, the physical entropy is defined based on a statistical ensemble P. The latter is very similar to an algorithmic model for x, except that in general a non-uniform probability distribution governs the elements of P, so the amount of information needed to specify an element $x' \in P$, on average, is given by Shannon entropy $H(P) = -\sum_{x'} P(x') \log P(x')$. Important paradoxes, such as the famous Maxwell's demon [25,27] or Gibbs' paradox [28], appear when it is realized that the ensemble P, and hence the entropy of the system, depends upon the knowledge d held by the agent, i.e., $P = P_d$. Such knowledge is usually given by macroscopic observations such as temperature, volume and pressure, and defines an ensemble P_d by the principle of maximal ignorance [29]. However, a more knowledgeable—or better equipped—agent may gather more information d' about the microstate, which in turn defines a more precise ensemble $P' \ni x$. This leads to incompatible measures of entropy. Zurek's physical entropy S_d includes the algorithmic information contained in d as an additional cost to the overall entropy measure of the system,

$$S_d = K(d) + H(P_d).$$

Note that the similarity with Equation (3) is not a mere coincidence. Zurek's physical complexity encompasses a two-part description of the microstate. It first describes a model—or an ensemble—for it, and second, it gives the residual information to obtain from the ensemble to the microstate, on average. In fact, when the ensemble takes a uniform distribution over all its possible elements, Shannon's entropy $H(P)$ reduces to the log-cardinality of the ensemble, which is, up to a $k_B \ln 2$ factor, Boltzmann's entropy.

With sufficient data d, the physical entropy S_d gets close to the complexity of the microstate $K(x)$. The ensemble P_d is then analogous to a sufficient statistics. Indeed, Baumeler and Wolf suggest [30] taking the minimal sufficient statistics as an objective—observer-independent—statistical ensemble (they call it *the* macrostate). Gell-Mann and Lloyd define [10] the complexity $K(P_d)$ of such a minimal sufficient ensemble to be the *Effective Complexity* of x. However, because of Vitányi's aforementioned equivalence between model classes, effective complexity is essentially the same idea as sophistication, which is why they also coincide numerically (see also [31] (lemma 21)). Finally, Müller and Szkola [31] have shown that strings of high effective complexity must have a very large logical depth, an idea to which we shall come back in Appendix A.

3. Defining Emergence

The previous discussion of the Kolmogorov structure function made manifest the fact that a drop like the one displayed in Figure 2 at complexity level α_M is associated with a distinguished model that accounts for meaningful properties of the data. In this example, all such properties were reflected in the description of S_M. In general, however, structure functions need not be characterized by a single drop. In this spirit, what should be thought of a string whose structure function has many drops, as displayed in Figure 3? With only a few bits of model, not much can be apprehended of x. With slightly more bits, there is a first model, S_1, capturing some useful properties of x, which leads to a more concise two-part description. Allowing even more bits, a second model S_2 is possible; while being more complex, this second model reflects more properties of x in such a way as to yield an even smaller two-part description. This series of models continues as the allowed complexity increases. Eventually, the structure function reaches the minimal sufficient statistics S_M, after which more complex models are of no help in capturing meaningful properties of x.
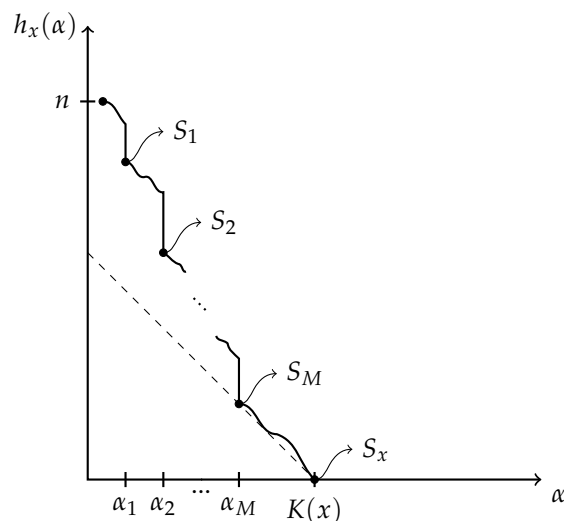

Figure 3. A structure function with many drops.

These observations illustrate that models can prove useful when not displaying all relevant properties of the data. Those "partial" models, while not sufficient, enable a most efficient description of the data with respect to all models of lower or equal complexity. Thus, in the same way that a model witnessing the minimal sufficient statistics is understood to capture the meaningful properties of the data, those intermediate models can be thought of as capturing only some of those meaningful properties. It is from this notion that the proposed definition of emergence is constructed.

Before going any further, a point needs to be addressed: Do strings with a structure function of many drops actually exist? Yes. In [32], it is shown that *all shapes are possible*, i.e., for any graph exhibiting the necessary properties mentioned in the previous section, there exists a string whose structure function lies within a logarithmic resolution of the graph.

The main idea of our proposal is to relate emergence to the phenomenon by which the experimental data x exhibits a structure function with many drops. They feature regularities that can be grasped at different levels of complexity.

3.1. Towards a Definition

In order to sharply define the models corresponding to drops of the structure function, and to make precise in which sense these are "new" and "understand" more properties, we construct a modified structure function upon which we formalize these notions. This shall take us to a definition of emergence.

3.1.1. Index Models

As discussed briefly in the previous section, one can construct models canonically from a given model by appending to it bits of the index of the data x in that model.

Definition 2. *(Index models) For a model $S \ni x$ and $i \in \{0, \ldots, \lceil \log |S| \rceil \}$, the index model $S[i]$ is given by the subset of S whose first i bits of index are the same as those of x.*

For concreteness, if $b_1 b_2 b_3 \ldots b_i$ are the first i bits of index of x, one way to compute $S[i]$ is to first execute the (self-delimiting) program that computes S, and then concatenate the following program:

$$\underbrace{\texttt{The following program has } (i+c) \texttt{ bits.}}_{K(i)+O(1) \text{ bits}} \tag{4}$$

$$\underbrace{\texttt{Among the strings of } S \texttt{, keep those whose index start with }}_{c \text{ bits}} b_1 b_2 b_3 \ldots b_i.$$

The first line of the routine is only for the sake of self-delimitation. Note that this concrete description of $S[i]$ implies

$$K(S[i]) \leq K(S) + i + K(i) + O(1).$$

Furthermore, for every bit of index given, the model $S[i] \ni x$ so defined contains half-fewer elements than S does. Hence,

$$\log|S[i]| = \log|S| - i.$$

As can be seen from the program displayed in Equation (4), because of self-delimitation, specifying i bits of index requires more than i extra bits of model description—it requires $\bar{\gamma}(i) := i + K(i) + c'$, where c' accounts for the constant-sized part of program (4). An inverse of $\bar{\gamma}(i)$ can be defined as

$$\bar{\imath}(\gamma) = \max_i \{i \colon i + K(i) + c' \leq \gamma\}, \tag{5}$$

which represents the number of index bits that can be specified with γ extra bits of model description. Denoting the index model $S(\gamma) := S[\bar{\imath}(\gamma)]$ and noticing that the difference between γ and $\bar{\imath}(\gamma)$ is of logarithmic magnitude, one finds that

$$h_x(K(S) + \gamma) \leq \log|S(\gamma)| = \log|S| - \bar{\imath}(\gamma) = \log|S| - \gamma + O(\log n).$$

3.1.2. A Modified Structure Function

We prescribe a procedure that associates to each complexity level α a model $S^{(\alpha)}$ which has log-cardinality very close to $h_x(\alpha)$. Intuitively, from lower to higher complexity, α is mapped to the witness of $h_x(\alpha)$ whenever α follows a large enough drop of the structure function; furthermore, whenever the structure function is in a slope -1 regime, α is mapped to an index model that is derived from the last witness of $h_x(\alpha)$.

Formally, let α_0 be the smallest complexity threshold for which $h_x(\alpha_0)$ is defined. The sequence of models $\{S^{(\alpha)}\}$ is defined recursively through

$$(S^{(\alpha_0)}, k_{\alpha_0}) = (S, \alpha_0) \text{ with } S \text{ the witness of } h_x(\alpha_0) \tag{6}$$

$$(S^{(\alpha)}, k_\alpha) = \begin{cases} (S, \alpha) \text{ with } S \text{ the witness of } h_x(\alpha) & \text{if } k_{\alpha-1} + h_x(k_{\alpha-1}) - \alpha - h_x(\alpha) \geq Q(k_{\alpha-1}) \\ (S^{(k_{\alpha-1})}(\alpha - k_{\alpha-1}), k_{\alpha-1}) & \text{otherwise}. \end{cases}$$

To determine whether a drop at complexity level α occurs or not, and so whether a new witness of the structure function is introduced or not, the length $\alpha + h_x(\alpha)$ of the two-part description at complexity level α is compared to that of the two-part description permitted by the previous model, $S^{(k_{\alpha-1})}$. If they differ by more than a threshold quantity $Q(k_{\alpha-1})$, a drop occurs. Alternatively, a drop can also be thought to occur if the structure function falls below the -1 slope regime by more than $Q(k_{\alpha-1})$. This quantity is given by

$$Q(\alpha) = K(h_x(\alpha)|\alpha) + O(\log \log n),$$

which, for $\alpha \geq K(n) + O(1)$, is smaller than $O(\log n)$ because $h_x(\alpha) \leq |S_{\text{Babel}}| = n$. As a final remark about Equation (6), a unique witness of $h_x(\alpha)$ is determined as the model that has $\log |S| \leq h_x(\alpha)$ and is produced by the first program of length α to halt with output S.

Definition 3. *(Modified structure function) The modified structure function $\tilde{h}_x(\alpha)$ is defined as*

$$\tilde{h}_x(\alpha) = \log|S^{(\alpha)}|.$$

It follows from this definition that $\tilde{h}_x$ lies within an additive logarithmic term above h_x, and the two functions coincide after a drop. Why define a modified structure function $\tilde{h}_x$, which is very close to the original structure function h_x? First, a "drop" of the structure function is clearly identified: it corresponds to a point in the construction of the modified structure function where the model used is updated rather than built with bits of index. Second, Equation (6) keeps track of the actual models used at each complexity threshold as witnesses of $\tilde{h}_x(\alpha)$. They are of only two kinds, either updated to a new witness of $h_x(\alpha)$ or built with more bits of index. In the original structure function, for two points α and β in a slope -1 regime, nothing guarantees that the models witnessing $h_x(\alpha)$ and $h_x(\beta)$ build on the same idea. They could a priori be completely different models, capturing completely different properties about the string x, but it just happens that the difference of their log-cardinality is roughly $\beta - \alpha$. However, the defining models of $\tilde{h}_x$ are constructed in a way that the -1 slope forces the models to reflect the same idea. They simply contain more or less of the index of x. It is the departure from the slope -1 regime in the function $\tilde{h}_x$ that indicates that a new model is used, one that intuitively captures other properties of x. These models shall be the ones of interest.

3.1.3. Minimal Partial Models as a Signature of Emergence

We have emphasized in the construction of the modified structure function a difference between the slope -1 regime and the drops of the structure function. Indeed, while the former amounts to index models, the latter corresponds to relevant yet partial models. These will be central to our proposed definition of emergence.

The set of numbers $\{k_\alpha\}_{\alpha \in \{\alpha_0, \dots, K(x) + O(1)\}}$ corresponds to the set of α's for which there are drops in the structure function.

Definition 4. *(Minimal partial models) The minimal partial models are defined as the witnesses of the drops of $\tilde{h}_x$, namely, the models $\{S^{(k_\alpha)}\}_{\alpha \in \{\alpha_0, \dots, K(x) + O(1)\}}$ as defined in (6).*

In what follows, we denote by $S_1, S_2, \dots, S_M$ the successive minimal partial models with respective complexity $\alpha_1 < \alpha_2 < \dots < \alpha_M$.

Definition 5. *(Emergence) Emergence is the phenomenon characterized by observation data that display several minimal partial models.*

It can be seen that the above definition maintains the generality expected of the notion of emergence, allowing for it to be applied in many different contexts. Moreover, it will be seen to allow for a mathematical treatment of various related notions.

In view of the comments in Section 1.2, emergence is a function of the observation string x and not necessarily of the real object that x is purported to represent. For instance, in the case of the dishonest scientist who disregards the object under investigation to give bits at whim, any emergence displayed by x would arise from the system that produced those bits—the scientist's brain, which is itself a part of reality.

3.2. Quantifying Emergence

Under the proposed definition of emergence, we develop quantitative statements. In this section, three theorems are presented. Ideas related to these results have previously been developed in algorithmic statistics [20,32,33].

To avoid many technicalities and precisions, the theorems presented in this section are formulated up to logarithmic error terms. More precise statements together with their corresponding proofs are given in Appendix A.

3.2.1. The Data Specifies the Minimal Partial Models

The first theorem confirms a basic intuition. The minimal partial models should be thought of as optimized ways to give the structural information about x, and so it should contain almost only information about x. The following theorem confirms that this is the case. Most of the algorithmic information of the minimal partial models is in fact information *about* x.

Theorem 1. *Each minimal partial model S_i can be computed from x and a logarithmic advice,*

$$K(S_i|x) = O(\log n).$$

Using the chain rule of Equation (2) entails rephrasing the statement as

$$K(x) = K(S_i) + K(x|S_i) + O(\log n),$$

so producing S_i in order to get x is not a waste; in fact, it is almost completely a part of the algorithmic information of x.

Proof. We give a program q of length $O(\log n)$ that computes S_i out of x.

```
q :  Give explicitly K(S_i) and ⌈log|S_i|⌉
     Run all programs p of length K(S_i) in parallel
     If p halts with U(p) = ⟨S⟩:
         If log|S| ≤ ⌈log|S_i|⌉ and x ∈ S:
             Print S and halt.
```

Only the first line of q has a non-constant length, and since both $K(S_i)$ and $\lceil \log |S_i| \rceil$ are smaller quantities than n, giving them explicitly requires $O(\log n)$ bits. □

3.2.2. Partial Understanding

We now justify the use of the term "partial" to qualify the non-sufficient minimal partial models. Intuitively, sharp drops of the structure function should be in correspondence with non-trivial properties of the underlying string. The minimal partial models at those points should encompass an "understanding" of these properties. Naturally, the magnitude

of this understanding could be equated to the size of the drop. Theorem 2 confirms this idea, when "understanding" holds the following meaning.

In the context of AIT, understanding amounts to reducing redundancy, as a good explanation is a simple rule that accounts for a substantial specification of the data. For instance, when one understands a grammar rule of some foreign language, that rule can be referred to in order to explain its many different instantiations. Those instantiations are redundant, and once the grammar rule is specified, this redundancy is reduced.

Definition 6. *(Redundancy) The redundancy of a string x of length n is defined to be*

$$\mathrm{Red}(x) := n - K(x|n).$$

The redundancy of a string is thus the number of bits of a string that are not irreducible algorithmic information. In other words, it is the compressible part of x. Redundancy could then be thought of as a quantification of how much there is to be understood about x upon learning x^*. Comparing x to x^*, however, is an all-or-nothing approach, and the purpose of non-probabilistic statistics is to make sense of partial understanding by studying (two-part) programs for x that interpolate between the "`Print x`" and the x^* explanations. The next definition generalizes redundancy so that it can be relative to an algorithmic model.

Definition 7. *(Randomness deficiency) The randomness deficiency of a string x, with respect to the model S is*

$$\delta(x|S) := \log|S| - K(x|S).$$

It measures how far x is from being a typical element of the set. Indeed, a typical element would have $K(x|S) = \log|S| + O(1)$ so that $\delta(x|S)$ essentially vanishes. Notice that the redundancy can be recovered from

$$\delta(x|S_{\mathrm{Babel}}) = n - K(x|S_{\mathrm{Babel}}) = \mathrm{Red}(x) + O(1).$$

We can then explore how much each minimal partial model reduces the randomness deficiency—or understands—the data x. Define d_i as the height of the drop just before getting to S_i, namely,

$$d_i := \tilde{h}_x(\alpha_i - 1) - \tilde{h}_x(\alpha_i).$$

Theorem 2. *The height of the i-th drop measures how much more S_i reduces the randomness deficiency, compared to S_{i-1}, i.e.,*

$$\delta(x|S_{i-1}) - \delta(x|S_i) = d_i + O(\log n).$$

Proof. Using the chain rule in Equation (2) twice, which amounts to a bayesian inversion, and Theorem 1,

$$
\begin{aligned}
\delta(x|S_i) &= \log|S_i| - K(x|S_i) \\
&= \tilde{h}_x(\alpha_i) - K(x) - K(S_i|x) + K(S_i) + O(\log n) \\
&= \tilde{h}_x(\alpha_i) - K(x) + \alpha_i + O(\log n).
\end{aligned}
\tag{7}
$$

With the help of Figure 4, and recalling that if γ extra bits of model description are available, $\bar{\imath}(\gamma) = \gamma + O(\log n)$ bits of index can be given, observe that

$$
\begin{aligned}
\tilde{h}_x(\alpha_{i-1}) - \tilde{h}_x(\alpha_i) &= \tilde{h}_x(\alpha_i - 1) - \tilde{h}_x(\alpha_i) + \tilde{h}_x(\alpha_{i-1}) - \tilde{h}_x(\alpha_i - 1) \\
&= d_i + \bar{\imath}(\alpha_i - 1 - \alpha_{i-1}) \\
&= d_i + \alpha_i - \alpha_{i-1} + O(\log n).
\end{aligned}
$$

Using Equation (7),

$$
\begin{aligned}
\delta(x|S_{i-1}) - \delta(x|S_i) &= \tilde{h}_x(\alpha_{i-1}) - \tilde{h}_x(\alpha_i) + \alpha_{i-1} - \alpha_i + O(\log n) \\
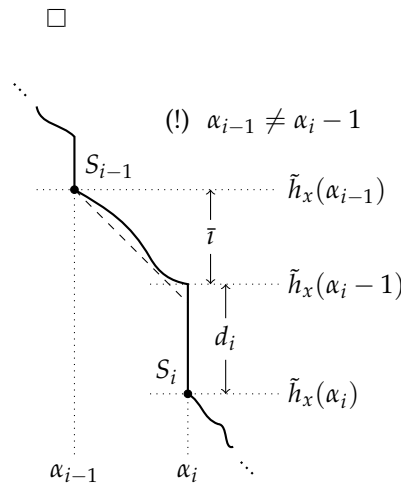&= d_i + O(\log n).
\end{aligned}
$$

□

Figure 4. A visual help for the proof of Theorem 2.

We can then interpret the algorithmic information in the minimal partial models as the algorithmic information of x that enables a reduction of the redundancy of x. This reduction of redundancy can be quantified by the sum of all previous drops, and the amount of redundancy left to be reduced is the sum of the drops to come. When the minimal sufficient statistic is described, with only α_M bits of the algorithmic information in x, the redundancy of x is completely reduced. The remaining information left to specify is then the index of $x \in S_M$, which is itself irreducible algorithmic information in x. However, this information is not the relevant structural information about x—it is the *incidental* information.

3.2.3. Hierarchy of Minimal Partial Models

The following theorem shows that the algorithmic information in the minimal partial models is organized in a nested structure, namely, the information in the simpler minimal partial models is mostly contained in the more complex ones. Put differently, the complex minimal partial models can be used to compute the simpler ones with some short advice.

Theorem 3. *For $j > i$,*
$$
K(S_i|S_j) \le \alpha_i - \alpha_{i-1} + O(\log n).
$$

The proof is relegated to Appendix A.

Note that when the structure function is very steep, the values $\{\alpha_i\}$ which mark the complexity of the minimal partial models in this steep regime are close to one another, so the leading term $\alpha_i - \alpha_{i-1}$ is small. In the limit where the (actual) structure function is so steep that it actually drops by more than $Q(\alpha_i - 1)$ between the two neighbouring points $\alpha_i - 1$ and α_i, then the leading term vanishes, and one finds $K(S_i|S_j) = O(\log n)$. This is formally stated and proved in the appendix (see Lemma A2).

3.3. Extending Concepts

We revisit the notions of coarse-graining and boundary conditions, broadening their scope.

3.3.1. A Notion of Coarse-Graining

Many approaches to emergence appeal to some notion of coarse-graining. For instance, the relevant quantities of a physical system might correspond to functions over state space. In this case, an important tool consists in averaging those quantities over regions of that space, retaining only the large scale structures, as is done in the method of effective field theories. In the context of algorithmic statistics, coarse-graining will be seen as a special case of what we will call *regraining*. We begin by defining the notion of coarse-graining precisely.

In set theory, coarse-grainings are defined from some mother set Ω. A *partition* $\mathcal{P}$ of Ω is a collection of disjoint and non-empty subsets such that their union gives back Ω. Let $\mathcal{P}_{\text{fine}}$ and $\mathcal{P}_{\text{coarse}}$ be two partitions of Ω. If every element in $\mathcal{P}_{\text{fine}}$ is a subset of some element of $\mathcal{P}_{\text{coarse}}$, then $\mathcal{P}_{\text{fine}}$ is a *refinement* of $\mathcal{P}_{\text{coarse}}$ and $\mathcal{P}_{\text{coarse}}$ is a *coarse-graining* of $\mathcal{P}_{\text{fine}}$. In physics, those partitions are usually specified through non-injective functions globally defined on the state space via the pre-images.

The key point of non-probabilistic statistics is to investigate an individual object x, without needing to refer to other x' in the set of bit strings. Hence, algorithmic models are disconnected from the notion of partition, since a single set is defined specifically for x, with no requirement to define a corresponding set for x'. As such, algorithmic models do not partition bit strings. Still, an algorithmic model $A \ni x$ could be qualified as a *model coarse-graining* of a $B \ni x$ if $B \subseteq A$. This type of model coarse-graining in fact occurs in the regime of index models. This motivates the extended notion of *regraining*, which is simply a change from some model $A \ni x$ to $B \ni x$, where neither model needs to be a subset of the other. It is qualified as a *fine* regraining if $|B| < |A|$ and a *coarse* one if $|B| > |A|$. Model coarse-grainings are particular cases of regrainings.

The *optimal regraining* corresponds to jumping along the minimal partial models. The optimal coarse regraining occurs in the direction from S_M to S_1, and corresponds to modelling fewer and fewer features of the data x to the benefit of having simpler and simpler models. It is optimal in the sense that this procedure will yield the best possible models over all complexity thresholds. As opposed to the usual approaches, where the coarse-graining occurs by averaging over specific variables (space, momentum, time...), our notion of regraining is parametrized by theory size, and so encompasses *all computable ways* in which a coarse-graining can be done. If properties of a physical system are such that the optimal coarse regraining happens by averaging over a specific variable, like space configuration, then the algorithmic regraining shall boil down to the usual method.

3.3.2. Boundary Conditions

Recall from Section 1.2 that an intrinsic difficulty of scientific investigation is that the recorded data x never perfectly reflects a single system. Even if we leave aside the effect of the measurement apparatus and the scientist on the data, it remains that systems are never completely isolated from an environment. As any interaction mediates an exchange of information, the effect of a large and complex environment will be modelled as random noise in models of small complexity. (An example of this situation is given by the dissipation–fluctuation theorem [34] that relates dissipative interactions in a system to the statistical fluctuations around its equilibrium point. Indeed, this theorem relates dissipation, an irreversible process that does not preserve information, with noise in the form of statistical fluctuations.) However, if the string x is sufficiently detailed, some structures of the environmental "noise" are grasped by models complex enough. This highlights that some information in x may be explained by models of great complexity but remain unexplained by a simple model. Such information can then be thought to reside outside of a simple model, namely, in its data-to-model code. This suggests that one should interpret the data-to-model code as the *boundary* of the model.

Definition 8. *(Boundary conditions) The boundary conditions of the model S corresponding to the data x are the index of x in S.*

In this definition, the scope of the term is broadened from its physical meaning, so that it can be thought of as the boundary of a model S, namely, the part of the system that generated the observational data x that is *not modelled* by S. The remaining structure in x is then viewed as coming from non-typical boundary conditions, often arising from interactions with an environment. In the case of the minimal sufficient statistics S_M, the typicality of x in S_M captures the fact that the boundary conditions are arbitrary with respect to the model.

The traditional space-time boundary conditions (e.g., the initial position and velocity of a particle) of a system are an example of what is usually relegated to the data-to-model code, as models usually do not aim at explaining them. Another example is the precise values of mechanical friction coefficients. Within classical mechanics, these values come from outside the theory and would thus be a part of the boundary conditions when understood as per Definition 8. However, with more precise observations, one could explain the values of the coefficients from a more precise model that encompasses molecular interactions. More examples are provided in the following section.

4. Applications

The versatility of the proposed approach to emergence is now illustrated through some applications. This section is not meant to be an exhaustive review of the possible uses of these definitions, but should rather be understood as a complement to the main exposition whose purpose is twofold. We first illustrate the ideas developed with a concocted example, then we connect some of them to dynamical systems in physics.

4.1. Simulation of a 2D Gas Toy Model

As a first application, we consider a toy model for a non-interacting 2D gas on a lattice. The gas is taken to be spatially confined on an $L \times L$ grid with a discrete time evolution. Using a pseudorandom number generator, we choose an initial position and momentum for each of the N particles. Each momentum is only a direction in the set $\{l, r, u, d\}$, corresponding to left, right, up and down. The gas then evolves according to simple rules. A single particle, represented by a 1 in the lattice, just keeps its trajectory and momentum, as in Figure 5. When it bounces off a boundary, its momentum gets flipped, as in Figure 6. Intersecting trajectories are represented as displayed in Figure 7.
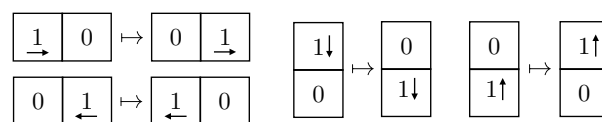

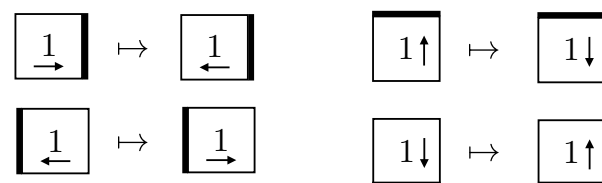**Figure 5.** A gas particle freely moving.

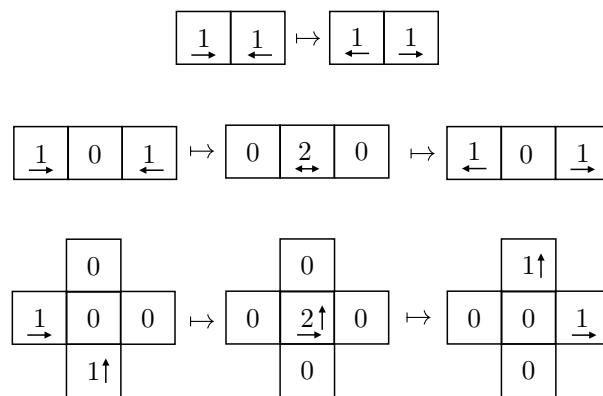**Figure 6.** A particle bouncing off walls.

Figure 7. Particles "collide" as if they go through one another.

At any time (including the initial time), if two or more particles are at the same site, we simply write down the number of particles in the site and keep track of the momenta. As an observation x, we extract for each of the first T time steps the state in configuration space (i.e., we ignore momentum). One visual way to encode the state in configuration space is to write in each of the L^2 sites a 0, and write to the left of it, in unary, the number of particles in that site. The table of 0s and 1s, can then be expanded into a string. For instance, a 3×3 grid example of this encoding is given in Figure 8.
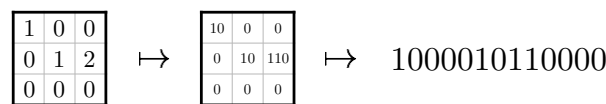

Figure 8. Encoding of the configuration state into bits.

At each time step, the bit string corresponding to the configuration state has one 0 for each site, and one 1 for each particle, so its total length is conserved in time and thus the observational data x could be obtained by a mere concatenation of the T configuration encodings. To have an idea of the course of x's structure function, models can be sought and used to determine upper bounds. In the case of this simulated 2D gas, x comes from the known context of the simulation, which provides important clues to find models other than the obvious S_{Babel} and $\{x\}$.

A first model inspired by the simulation specifies the parameters L and N, external to the gas, together with the final time T. Compared to $\{x\}$, which lists everything about the simulation, simplicity is gained by leaving outside of the model the initial conditions of the gas. This defines the set $S_{\text{Gas}}^{L,N,T}$ of all configuration histories of T iterations, for each possible initial conditions of N particles confined to a $L \times L$ grid. The size of the shortest program for $S_{\text{Gas}}^{L,N,T}$ amounts to $K(L,N,T) + O(1)$, since the evolution rules are of constant length. Different elements of $S_{\text{Gas}}^{L,N,T}$ are similar gases in different initial conditions, according to the identification of the index of $x \in S_{\text{Gas}}^{L,N,T}$ with the boundary conditions.

Even simpler models can be made by pushing into the boundaries the particular values of the external parameters L, N or T. For illustration, we make the argument only for T. Let T be expressed by τ bits in a binary expansion. A simpler model than $S_{\text{Gas}}^{L,N,T}$ can be given by producing, for each possible initial condition, *all* histories of length smaller than 2^τ. We denote this set as $S_{\text{Gas}}^{L,N,<2^\tau}$. Its cardinality is $2^\tau - 1$ times bigger than $S^{L,N,T}$, thus adding τ to the log-cardinality axis. If T was a random number,

$$K(T) = \tau + K(\tau) + O(1),\tag{8}$$

and exactly τ bits would be saved on the complexity axis, since only τ, as opposed to T, would be needed to compute the model. In general, however, T is not algorithmically random—it may also contain structures. Since $S_{\text{Gas}}^{L,N,<2^\tau}$ only encodes τ, a basic upper bound

for T, most of the information about T lies outside of what is modelled; it is again the index of x that carries this information as boundary conditions with respect to that model. In a similar fashion, other simple models can be introduced for L and N, pushing again their information into the boundaries of the models.

The models $S_{\text{Gas}}^{L,N,T}$ and $S_{\text{Gas}}^{L,N,<2^{\tau}}$ provide upper bounds to the structure function. Yet, as presented in Figure 9, upper bounds to the structure function still leave one uncertain about the actual course of the structure function. In particular, in the simulation of the gas, the initial conditions were not algorithmically random, as they came from a pseudorandom number generator—a short program which, on a short inputted seed, outputs a sufficiently long string. Hence, the program in the random number generator, together with its seed, is shorter than the length of the initial conditions it generated. The actual structure function reflects this through one more drop at a higher level of complexity. The hypothetical witness of this drop, S_{RNG}, is a model that explains the initial conditions as coming from the pseudorandom number generator. It is the set of all gas histories compatible with the dynamics previously described, and where the initial conditions have been generated with the pseudorandom program, with the seed being relegated to the data-to-model code. If the slope after S_{RNG} remains in a slope -1 regime, the seed is typical. However, in reality, the seed has been produced by another physical system, for instance, by the programmer. Yet again, if the seed is long enough, the structure function could reveal more drops which capture more structures, for instance, the favourite keys of the programmer who entered the seed on his keyboard. This process will go on until all that can be explained has been explained.
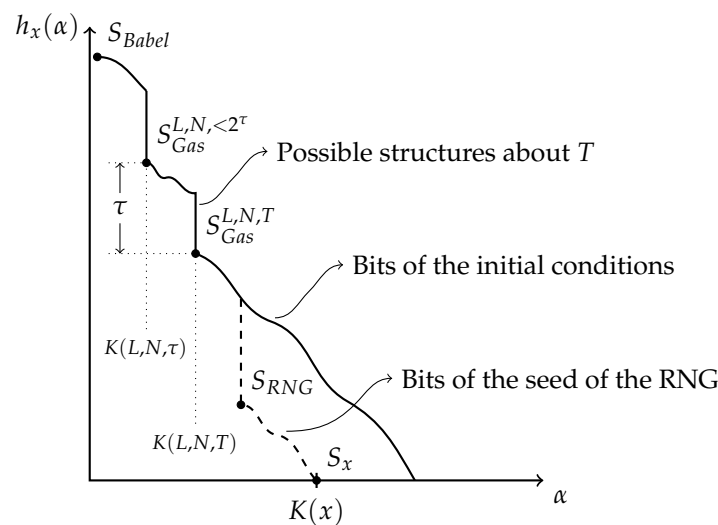

Figure 9. The solid line represents the known upper bound of the structure function. The dashed line represents the hypothesized real structure function.

This example makes clear that the notion of boundary conditions really refers to a theory (or an algorithmic model), and they are fixed somewhat arbitrarily, when the users of the theory are satisfied with their notion of the system that is being modelled. In this case, if what we want to model is the gas, then $S_{\text{Gas}}^{L,N,T}$ is good enough, and it is practical to declare that the initial state is typical. However, the reality may be quite different, and what we prescribe as a boundary condition to our theory may in fact be explained by a more complex, deeper theory.

4.2. Dynamical Systems

In this second application, we investigate how the notions introduced in this paper appear in the context of dynamical systems. We begin by documenting how the concept of integrability and chaos can be cast in the language of algorithmic information theory.

This is followed by an account of how thermodynamics can be seen to emerge, under the proposed definition of emergence, from the application of statistical mechanics to complex dynamical systems.

4.2.1. From Integrability to Chaos

Consider a generic classical system with Hamiltonian H and where the state space M is indexed by a set of real coordinates $X = \{q_i, p_i\}_{i \in \{1,\dots,\dim M/2\}} \in M$. (More precisely, M is a symplectic manifold parametrized locally by real coordinates forming an atlas.) Solutions to the dynamics are curves in M describing the evolution of the state in time. Specifying M, H and an initial point $X_0 \in M$ singles out a unique solution curve X_t of the dynamics. As a rudimentary formalization of some observation of the system, consider a bounded observable represented by a function f with $f : M \to [0,1]$. A discrete sequence is constructed from its evaluation $\{f(X_{j\tau})\}_{j \in \{1,\dots,N\}}$ at a regular time interval $0 < \tau \in \mathbb{Q}$ with negligible $K(\tau)$. As this sequence is to represent a series of measurements, one must restrict its resolution, since measured values are always constrained to a finite resolution. For a real number α, we denote by $[\alpha]_k$ the truncation of its binary expansion after the first k bits beyond the decimal point such that $|[\alpha]_k - \alpha| \le 2^{-k}$. This truncation effectively restricts the resolution to k bits as the measurement function is upper-bounded by 1. Denoting by $f_j^k \equiv [f(X_{j\tau})]_k$ the restricted measurements, the recorded observational data string x is then an encoding of the sequence of measurements:

$$x \equiv \langle \{f_j^k\}_{j \in \{1,\dots,N\}} \rangle.$$

We now wish to characterize the complexity of the data string x and study its asymptotic behaviour when the length N of the measurement sequence is increasing. First, one must formulate a meaningful upper bound for $K(x)$. To that end, we require f to preserve information, that is,

$$K([f(X)]_k \,|\, [X]_k) = O(1).$$

A trivial bound is then given by the bit length of the encoded sequence of measurements; thus,

$$K(x) \le kN + O(\log kN).$$

However, the regularity provided by the laws of motion implies that this bound is not strict. Indeed, given the Hamiltonian H and the manifold M, the machinery of symplectic geometry specifies the dynamical evolution as a set of differential equations that we will denote as $\langle H, M \rangle$. These equations can be integrated numerically from the initial conditions X_0 to obtain f_j to a desired precision. These remarks, together with the stated condition on f, imply that

$$K(x) \le K(\langle M, H \rangle, \tau, k, N, X_0) + O(1).$$

The above can be further simplified in view of studying the asymptotic behaviour in N by observing that the dynamical laws $\langle M, H \rangle$, the time interval τ and the resolution k are fixed and independent of N. Thus, as the length of x is scaled by increasing N, they can be taken to be constant. (To simplify the analysis, it is tacitly assumed that the dynamical laws are simple in the sense that the coefficients of the differential equations in $\langle H, M \rangle$ are of finite complexity.) Hence, one has

$$K(x) \le K(N) + K(X_0) + O(1). \tag{9}$$

Remembering that X_0 encodes the initial conditions, which are a set of real numbers that cannot be constructively specified in general, one is left with a conundrum. Indeed, if X_0 encodes typical real numbers, the upper bound (9) is trivial as the right-hand side is infinite. However, only a finite precision in the initial conditions is required in order to integrate the system to a given precision in the final result. Thus, the resolution in X_0 required is only as

much as is needed to compute $\{f_j^k\}_{j\in\{0,1,...,N\}}$. As such, the asymptotic behaviour of $K(x)$ for $N \to \infty$ is determined by the scaling in the required resolution.

A chaotic dynamical system is often characterized by an exponential divergence in the evolution of nearby initial configurations, namely,

$$\frac{|X_t' - X_t|}{|X_0' - X_0|} = e^{\lambda t},$$

where $|\cdot|$ denotes a metric on M and λ is known as the Lyapunov exponent. (It is here assumed that the Lyapunov exponent is constant and unique, which is not always the case.) In such a chaotic system,

$$|X_0' - X_0| < 2^{-\lambda' j - k} \qquad \Longrightarrow \qquad |X_{j\tau}' - X_{j\tau}| < 2^{-k}, \qquad \text{with} \qquad \lambda' = \frac{\lambda \tau}{\ln 2},$$

so k bits of precision on $X_{j\tau}$ can be achieved by $k + \lambda' j$ bits of precision on X_0. Therefore, the computation of $X_{N\tau}$ from the initial condition is more efficient—in terms of description length—than straightforward enumeration if $k + \lambda' N \le kN$, or equivalently, if

$$\lambda' \le k - \frac{k}{N}. \tag{10}$$

This means that for some values of the Lyapunov exponent λ, time interval τ and precision k, it could be more efficient to simply recite the observed data $\{f_j^k\}_{j\in\{1,...,N\}}$ as a genuinely random string. However, no matter how large the Lyapunov exponent is, there are time steps τ small enough to make it more efficient to calculate $\{f_j^k\}_{j\in\{1,...,N\}}$ from enough bits of initial conditions. Concretely, the precision on the initial conditions that can be obtained is bounded by the resolution of measurement devices. A more practical approach accounts for this with a fixed resolution $k' > k$ in the initial conditions and is thus limited to the truncation $[X_0]_{k'}$. This, together with the Lyapunov exponent of the system under consideration, determines an interval of predictability within which the observational data x can be compressed. To preserve predictability beyond this interval, one is forced to update one's knowledge of the state of the system with a measurement. The phenomenon is well-known within chaos theory and shows up as a fundamental limitation to the predictability of such systems. Seen from the algorithmic information lens, chaos can be understood as follows: the bits of the state's description that are initially far away from the decimal point, and hence apparently irrelevant, make their way closer to the decimal point, becoming relevant.

Dynamical systems can generally be organized by considering the asymptotic of the string of measurements x with $N \to \infty$. At one end of the spectrum lie integrable systems, where k bits of knowledge of X_0 can be used all the way through to compute the k bits of f_N^k. Those include systems where integration can be carried symbolically without an accumulation of errors. On the other side of this spectrum are chaotic systems, where $k + \lambda' N$ bits of X_0 are required to compute the k bits of f_N^k. Similar classification schemes for dynamical systems that account for integrability and the appearance of chaos based on computational complexity have been proposed previously [35]. An algorithmic perspective on dynamical systems brings the possibility of considering other types of systems, where $k + g(N)$ bits of X_0 can be used to compute the k bits of f_N^k, with $g(N)$ some a priori generic function.

4.2.2. Thermodynamics and Statistical Mechanics

Statistical mechanics posits the ergodicity of a complex dynamical system in order to obtain a partial, yet useful, description of its behaviour. This partial description is mostly understood to refer to the macroscopic description of a system displaying intractable microscopic descriptions. The generic approach is as follows. Starting again with a Hamiltonian and the associated phase space M, one first investigates the quantities conserved by the

time evolution. By fixing those conserved quantities, one establishes constraints that restrict the dimensionality of the accessible phase space. Properly defined, those constraints effectively foliate the phase space into a family of submanifolds $F \subseteq M$ that are each preserved by time evolution. The ergodic hypothesis now posits that the curves X_t produced by an initial point $X_0 \in F$ under the time evolution are dense in each submanifold F such that the time average value of an observable $\mathcal{O} : M \to \mathbb{R}$, over such a curve is equal to the average of the same quantity over a uniform measure on each submanifold F,

$$\lim_{T \to \infty} \frac{1}{T} \int_{t_0}^{t_0+T} \mathcal{O}(X_t) dt = \int_{F_{X_0} \subseteq M} \mathcal{O}(X) d\mu(X),$$

for μ the uniform measure over F. This uniform measure over the submanifolds F is often specified indirectly in terms of the Boltzmann weights of a state $X \in M$ over the submanifolds. With the above in mind, thermodynamics can be seen as the study of the interrelation of a relevant collection of macroscopic observables $\{\mathcal{O}_i\}$, expressing the change in the value of some observables in terms of the change in the value of the others. Such a thermodynamic description of a complex system is partial yet useful and relevant to the scale at which one would like to investigate the system.

Let us now concentrate on how this very generic picture of statistical mechanics and its relation to thermodynamics fits under our proposed definition of emergence. We first define the truncation $[F]_k$ of a submanifold $F \subset M$ to resolution k as the truncation of all coordinates of F to a k-bit resolution. (Technically this truncation depends on the chart, but we take an encoding of F to include an atlas and one for $[F]_k$ to include a prescription on the choice of the chart in which to truncate each points.) Then, positing the ergodicity of the system under study enables a direct reframing of statistical mechanics in terms of the ideas of this paper. Indeed, the postulated uniform measure on submanifolds of the phase space M amounts to postulating the corresponding microscopic states in a submanifold to be equally likely under time evolution. In other words, for some large enough finite time interval τ, the sequence

$$x_N \equiv \langle \{ [X_{n\tau}^{(i)}]_k \}_{i \in \{1,...,\dim F\}, \, n \in \{1,...,N\}} \rangle, \quad \text{for} \quad X^{(i)}{}_{i \in \{1,...,\dim F\}} \quad \text{coordinates on F,}$$

is a typical sample of the truncated submanifold $[F]_k$. The lower bound on the time interval τ that needs to be satisfied for the above to hold is related to the Lyapunov exponent of the system. Indeed, such a bound corresponds to time intervals satisfying the converse of Equation (10). In such a case, x_N is essentially an algorithmically random string. From this observation, it follows that for a sufficiently large time interval,

$$K(x_N) = K([F]_k) + N \log(|[F]_k|), \tag{11}$$

which indicates that the model $[F]_k$ for the string x_N is an algorithmic sufficient statistics.

The above discussion emphasized how thermodynamics, together with the ergodic hypothesis, amount to postulating that the models in Equation (11) associated with the decomposition into invariant submanifolds are sufficient. Indeed, a thermodynamical description of the system at equilibrium is in correspondence with such a decomposition of the phase space, provided that the conserved quantities that define the submanifolds are taken to be the thermodynamical variables. Note that $[F]_k$ is not a sufficient statistics if and only if the ergodic hypothesis fails. In such a case, one may still consider $[F]_k$ as a model of the data, but then the time sampling of the system shall not be algorithmically random; more structures can be found and incorporated to thermodynamical model.

5. Conclusions

We proposed a mathematical and objective definition of emergence cast in the language of algorithmic information theory. Yielding clear and far-reaching incompleteness

results [36,37], this field is rich enough to mathematize mathematics itself. In this paper, we used algorithmic information theory to mathematize epistemology, giving rise to a framework which encapsulates emergence.

Intuitively, emergence is the appearance of novel properties exhibited by a complex system. In most discussions about emergence, the criteria of novelty highly depend upon the field: the aerodynamicist may be stunned by new patterns in fluid dynamics; the biochemist, by new ways in which enzymatic networks interact. In our proposed definition, emergence occurs in *"theory space"*: the thresholds of emergence are marked by the complexity of models that enable an overall shorter expression of the observed data. These models can be thought of as *understanding new structures*. Although the considered models seem very constrained (they are sets of finite bit strings), they are in fact as general as they can be since they are rooted in universal computation: any "new pattern in fluid dynamics" or "enzymatic networks interaction" that can be described is amenable to a computational process and thus an algorithmic model.

The development of our proposal was done through the minimal partial models. In Section 3, we proved that:

1. The data specifies almost everything about the minimal partial models;
2. The magnitude of the drop measures the amount of new understanding;
3. More complex minimal partial models specify almost completely the simpler ones.

We also extended the notions of coarse-grainings and boundary conditions, freeing them from any specific theory. In Section 4, we considered some applications to dynamical systems and thermodynamics, and found that a new light can be shed on chaos and on the ergodic hypothesis.

The absolute generality of algorithmic-information-theoretic methods comes at the price of uncomputability. For instance, the shapes of Figure 9, in Section 4.1, are only conjectured. No program can return the structure function of a piece of data x. Nevertheless, the definition provides a precise framework to discuss the notion of emergence. A relaxation to the context of limited computational resources may be of interest in order to find concrete utility and applications in real-life computations; while some of the results obtained might no longer hold, the definition itself can still be applied within this limited computational context.

We recognize that the concepts involved in the proof of Theorem 3 in the upcoming Appendix A.3 challenge the reconciliation between our mathematical proposal and the youth of our universe. For a long string, the deep models, namely, those that occur at late drops of the structure function, are the result of programs that terminate after an unthinkably long computation. They have the largest finite running times among all programs no larger in size, so they solve the halting problem for shorter programs. This is the busy beaver regime. At a mere 14 billion years old, our universe seems too young to accommodate such computations, and this seems to hold even if we take into account the parallel computation that occurs in the observable universe. Indeed, as Bennett once put it, "the cube of Hubble's length over Plank's length is not even breakfast for the busy beaver!" However, assessing the actual computational capabilities of all physical phenomena in the entire universe is an open problem. Another way out of this conundrum is to leave the deep information in the initial conditions of the universe. One could then ask about the source of this information. Yet another possibility is that systems in nature are confined to relatively shallow models.

Facing the realization that models witnessing drops of the structure functions are made of halting information, Vereshchagin and Shen [20] wrote "This looks like a failure. [...] [I]f we start with two old recordings, we may get the same information [about their minimal sufficient statistic], which is not what we expect from a restoration procedure. Of course, there is still a chance that some Ω-number [halting information] was recorded and therefore the restoration process indeed should provide the information about it, but this looks like a very special case that hardly should happen for any practical situation." Facing this, they suggest considering models of more restricted classes or adding some additional

conditions theirby looking for "strong models". On the contrary, we think that that the minimal sufficient statistics of two recordings *should* share information, as they inevitably share a very common origin, which the model aims to capture. That this shared information is about the halting problem simply reflects the fact that their plausible common origin is the fruit of a very long computation, and not that the recording has anything to do with an Ω-number, or any direct representation of the halting problem.

Let us conclude with philosophy of science. Consider the string x to be an encoding of all scientific data ever recorded. Scientific theories aim at explaining x by grasping patterns in the data in order to reduce its redundancy. They are expressed in terms of models that distillate the structures from the apparently noisy boundary conditions. In an effort to tell apart theories or through pure experimental curiosity, the data x always increases. Through conjectures, scientists suggest new models that can better explain the observations. Better theories arise; they find important structures in apparent noise, or unify different models under the same umbrella. These models can either be proven false by some eventually contradicting piece of data, or discarded when a simpler and fitter model is found, namely, one that sits closer to the structure function. *Hence the process of scientific investigation may be identified with the upper semi-computation of the structure function of our observations*. The computational impossibility of deriving the optimal models from the data is reminiscent of the non-existent scientific induction. Theories are conjectured as tentative best explanations: they can never be proven right, nor optimal, nor final, as mandated by *fallibilism*.

Appendix A. Technical Precisions

This appendix expands the three theorems of Section 3.2 in much more detail, and with more precision.

Appendix A.1. Precisions on the First Theorem

Theorem A1. *Each minimal partial model S_i can be computed from x^* and a logarithmic advice of precise length $K(K(S_i), \lceil \log |S_i| \rceil \mid x^*) + O(1)$, and no shorter algorithm exists. In other words,*

$$K(S_i|x^*) = K(K(S_i), \lceil \log |S_i| \rceil \mid x^*) + O(1).$$

Proof. We first show the "$\leq$" part of the above relation using a method similar to that which was used in the proof of Theorem 1, namely, we give a program q of length $K(K(S_i), \lceil \log |S_i| \rceil \mid x^*) + O(1)$ that computes S_i out of x^*.

$$q: \quad \texttt{Compute } K(S_i) \texttt{ and } \lceil\log|S_i|\rceil \texttt{ from } x^*$$
$$\texttt{Run all programs of length } K(S_i) \texttt{ in parallel}$$
$$\texttt{If } p \texttt{ halts with } \mathcal{U}(p) = \langle S \rangle\texttt{:}$$
$$\texttt{If } \log|S| \leq \lceil\log|S_i|\rceil \texttt{ and } \mathcal{U}(x^*) = x \in S\texttt{:}$$
$$\texttt{Print } S \texttt{ and halt.}$$

To prove the "$\geq$" part of the relation, we invoke the following property [38] (Theorem 69):

$$K(S_i, x) = K(S_i, x, K(S_i)) + O(1),$$

and because $\lceil \log |S_i| \rceil$ is computable from S_i with a $O(1)$ advice,

$$K(S_i, x) = K(S_i, x, K(S_i), \lceil \log |S_i| \rceil) + O(1).$$

Making use of the chain rule (1) and subtracting $K(x)$ on both sides,

$$K(S_i | x^*) = K(S_i, K(S_i), \lceil \log |S_i| \rceil | x^*) + O(1).$$

The conclusion follows from

$$K(S_i, K(S_i), \lceil \log |S_i| \rceil | x^*) \geq K(K(S_i), \lceil \log |S_i| \rceil | x^*) + O(1).$$

$\square$

Appendix A.2. Precisions on the Second Theorem

The following precision of Theorem 2 leverages an alternative way of defining the randomness deficiency, which, because of its conditioning on S^* rather than S, favours the precise chain rule Equation (1).

Definition A1. *(Strong randomness deficiency) The strong randomness deficiency of a string x, with respect to the program S^* for the model S^*, is*

$$\delta(x | S^*) := \log |S| - K(x | S^*).$$

Moreover, a very slow $\mathbb{N} \to \mathbb{N}$ function is invoked, the $\log^*(n)$, defined as the number of times we can iterate taking the binary logarithm with a positive result, starting from n. For instance, $\log^*(17) = 4$ because $\log \log \log \log \log 17 < 0 < \log \log \log \log 17$.

A very tight upper bound to $K(x)$ is $\ell^*(x) + O(1)$, defined as

$$\ell^*(x) := \log^*(x) + l(x) + l(l(x)) + l^{(3)}x + \dots, \tag{A1}$$

where the sum is taken over all positive terms, and $l(x) = |x|$ alternatively denotes the length of x (see [11], Equation (3.2)).

Before revisiting Theorem 2, the following lemma quantifies with more precision the discrepancy between γ and $\bar{\imath}(\gamma) = \max_i\{i: i + K(i) + c' \leq \gamma\}$, previously defined in Equation (5).

Lemma A1.

$$\bar{\imath}(\gamma) = \gamma - K(\gamma) + O(\log^* \gamma).$$

Proof. The function $\bar{\imath}(\gamma)$ was motivated as an inverse to the function $\bar{\gamma}(i) = i + K(i) + c'$. This is rightly justified since

$$\bar{\gamma}(\bar{\imath}(\gamma)) = \gamma + O(1).$$

Indeed, by definition, $\bar{\imath}(\gamma)$ satisfies the constraint $\bar{\imath}(\gamma) + K(\bar{\imath}(\gamma)) + c' \leq \gamma$, i.e., $\bar{\gamma}(\bar{\imath}(\gamma)) \leq \gamma$. On the other hand, if $\bar{\gamma}(\bar{\imath}(\gamma))$ does not lie only within a $O(1)$ term below γ, then $\bar{\imath}(\gamma)$ could not be the maximal i such that $i + K(i) + c' \leq \gamma$. Therefore, $\gamma - \bar{\imath}(\gamma) = K(\bar{\imath}(\gamma)) + O(1)$.

It remains to show that $K(\bar{\imath}(\gamma)) = K(\gamma) + O(\log^* \gamma)$. Hereinafter, we denote i as a shorthand for $\bar{\imath}(\gamma)$ and *the displayed equations hold up to an $O(1)$ error term*. Observe that

$$K(i) = K(i, K(i)) = K(\gamma, K(i)) = K(\gamma) + K(K(i)|\gamma^*).$$

Bounding further the error term $K(K(i)|\gamma^*) := K^{(2)}(i|\gamma^*)$,

$$
\begin{aligned}
K(K(i)|\gamma^*) &= K(\, K(\gamma, K(i)) \,|\gamma^*) \\
&= K(\, [K(\gamma) + K(K(i)|\gamma^*)] \,|\gamma^*) \\
&= K(\, K(K(i)|\gamma^*) \,|\gamma^*) + O(3) && := K^{(3)}(i|\gamma^*) + O(3) \\
&= K(\, K(K(\gamma, K(i))|\gamma^*) \,|\gamma^*) \\
&= K(\, K([K(\gamma) + K(K(i)|\gamma^*)] \,|\gamma^*)|\gamma^*) \\
&= K(\, K(K(K(i)|\gamma^*)|\gamma^*) \,|\gamma^*) + O(4) && := K^{(4)}(i|\gamma^*) + O(4) \\
&= \ldots \\
&= K^{(m)}(i|\gamma^*) + O(m).
\end{aligned}
$$

Generally, $y \leq b \implies K(y|b) \leq |b| + O(1) = \log b + O(1)$, because

$$
\begin{aligned}
K(y|b) &\leq K(y||b|) \\
&\leq K(|b| - |y| \mid |b|) + K(y||y|) \\
&\leq K(|b| - |y|) + |y| \\
&\leq |b| - |y| + |y|.
\end{aligned}
$$

Since i and γ differ by a logarithmic term, they have the same length, plus or minus 1; hence, $|i| = |\gamma| + O(1)$. Moreover, $\ell^*(i)$ can be computed from $|i|$, so

$$K(K(i)|\gamma^*) \leq K(K(i)|\ell^*(i)) \leq \log \ell^*(i).$$

In a similar fashion,

$$K^{(3)}(i|\gamma^*) = K(\, K(K(i)|\gamma^*) \,|\gamma^*) \leq K(\, K(K(i)|\gamma^*) \,|\log(\ell^*(i))) \leq \log\log(\ell^*(i)),$$

and iteratively,

$$K^{(m)}(i|\gamma^*) \leq \log^{(m-1)}(\ell^*(i)) \leq \log^{(m-1)}(i).$$

Once m is such that $m = \log^*(i)$, $\log^{(m-1)}(i) \leq 2 = O(1)$. Hence, $\bar{\imath}(\gamma) = \gamma - K(\gamma) + O(\log^* \gamma)$.
$\square$

Theorem A2. *The height of the i-th drop measures how much more S_i^* reduces the alternative randomness deficiency, compared to S_{i-1}^*; i.e.,*

$$\delta(x|S_{i-1}^*) - \delta(x|S_i^*) =$$
$$d_i + \underbrace{K(\alpha_i, \lceil \log|S_i| \rceil \,|\, x^*) - K(\alpha_{i-1}, \lceil \log|S_{i-1}| \rceil \,|\, x^*) - K(\alpha_i - \alpha_{i-1})}_{\text{Explicit } O(\log n) \text{ quantity}}$$
$$+ O(\log^*(\alpha_i - \alpha_{i-1})).$$

Proof. Using the chain rule in Equation (1) twice, which amounts to a Bayesian inversion, and Theorem A1,

$$\begin{aligned}
\delta(x|S_i^*) &= \log|S_i| - K(x|S_i^*) \\
&= \tilde{h}_x(\alpha_i) - K(x) + K(S_i) - K(S_i|x^*) + O(1) \\
&= \tilde{h}_x(\alpha_i) - K(x) + \alpha_i - K(K(S_i), \lceil \log|S_i| \rceil \,|\, x^*) + O(1).
\end{aligned} \tag{A2}$$

Lemma A1 then yields (a glance at Figure 4 can be helpful)

$$\begin{aligned}
\tilde{h}_x(\alpha_{i-1}) - \tilde{h}_x(\alpha_i) &= \tilde{h}_x(\alpha_i - 1) - \tilde{h}_x(\alpha_i) + \tilde{h}_x(\alpha_{i-1}) - \tilde{h}_x(\alpha_i - 1) \\
&= d_i + \bar{\imath}(\alpha_i - 1 - \alpha_{i-1}) \\
&= d_i + \alpha_i - \alpha_{i-1} - K(\alpha_i - \alpha_{i-1}) + O(\log^*(\alpha_i - \alpha_{i-1})).
\end{aligned}$$

Using Equation (A2) and the above,

$$\begin{aligned}
\delta(x|S_{i-1}^*) - \delta(x|S_i^*) &= \tilde{h}_x(\alpha_{i-1}) - \tilde{h}_x(\alpha_i) + \alpha_{i-1} - \alpha_i - K(K(S_{i-1}), \lceil \log|S_{i-1}| \rceil \,|\, x^*) \\
&\quad + K(K(S_i), \lceil \log|S_i| \rceil \,|\, x^*) + O(1) \\
&= \alpha_i - \alpha_{i-1} - K(\alpha_i - \alpha_{i-1}) + d_i + O(\log^*(\alpha_i - \alpha_{i-1})) + \alpha_{i-1} - \alpha_i \\
&\quad - K(K(S_{i-1}), \lceil \log|S_{i-1}| \rceil \,|\, x^*) + K(K(S_i), \lceil \log|S_i| \rceil \,|\, x^*) + O(1) \\
&= d_i + K(\alpha_i, \lceil \log|S_i| \rceil \,|\, x^*) - K(\alpha_{i-1}, \lceil \log|S_{i-1}| \rceil \,|\, x^*) - K(\alpha_i - \alpha_{i-1}) \\
&\quad + O(\log^*(\alpha_i - \alpha_{i-1})).
\end{aligned}$$

$\square$

Appendix A.3. Precisions on the Third Theorem and Its Proof

The proof of Theorem 3 hinges on an unexpected relation between the structure function of the string x and an a priori completely different characterization of the patterns in x which invoke the running time of programs that compute it. This last notion is closely related to Bennett's *logical depth* [39], and the relation between it and the structure function has been established in [22] and further detailed in [20]. For the sake of completeness and precision, a self-contained proof is given here.

In computability theory, the unobtainable knowledge about the halting problem is usually represented as an infinite bit string which plays the role of an oracle. In algorithmic information theory, however, finite pieces of halting information can be represented finitely. One such representation is given by the first values of the busy beaver function. We denote the running time of a program p by $\text{RT}(p)$ and the fact that the universal Turing machine halts on input p by $\mathcal{U}(p) \searrow$.

Definition A2. *(Busy beaver function) The busy beaver is a function $B : \mathbb{N} \to \mathbb{N}$ defined by*

$$B(i) := \max \{ \text{RT}(p) : \mathcal{U}(p) \searrow \text{ and } |p| \leq i \}.$$

In words, it is the maximal finite running time of a program of i bits or less. The witness of $B(i)$—the last program of length i or less that halts—shall be denoted $\bar{q}_i$.

A program p is a (τ, ℓ)-*program* for x if $\mathcal{U}(p) = x$, $|p| = \ell$ and $B(\tau - 1) < RT(p) \leq B(\tau)$. The latter condition means that there is a τ-bit halting program that runs for at least as long as p runs, but none of length $\tau - 1$ or less. Out of all (τ, ℓ)-programs, of interest are the ones of minimal ℓ for a given τ, or vice versa. For a string x, the *time profile* is thus defined as the boundary of the region

$$\{(\tau', \ell') : \tau' \geq \tau, \ell' \geq \ell, \exists (\tau, \ell)\text{-program for } x\}.$$

In the spirit of non-probabilistic statistics and their connection with two-part descriptions (see Section 2.2), the model S is a (α, m)-*two-part description* for x if $x \in S$, $K(S) = \alpha$ and $K(S) + \log |S| = m$. Out of all two-part descriptions, the optimal ones are those of minimal m for a given α—they are recognized as the witnesses of the structure function. The *description profile* is thus defined as the graph $(\alpha, \alpha + h_x(\alpha))$, and analogously with the time profile, this graph corresponds to the boundary of all (α, m)-descriptions.

Theorem A3. *For $j > i$,*

$$K(S_i|S_j) \leq \alpha_i - \alpha_{i-1} + 3K(\alpha_{i-1}) + O(\log \log n).$$

Proof idea. It shall first be established that the time profile and the description profile are close to one another. This means that the running time of the two-part descriptions of the minimal partial models is of the busy beaver magnitude. Since most of the computing time happens in the execution of the models—and not in the data-to-model code—their slow running time makes them full of halting information. The later minimal partial models then have most of the information of the prior ones.

Proof. The upcoming two propositions establish that for any data string x, what a priori looks like two different ways of characterizing the structure in x, namely, the time profile and the description profile, are in fact closely related to one another. As a set of points on the plane, they are close to one another. Again for this proof, *the displayed equations hold up to an $O(1)$ error term.* $\square$

Proposition A1. $\exists a \ (\alpha, m)$-*two-part description* $\implies \exists a \ (\tau, \ell)$-*program, with $\tau \leq \alpha + O(1)$ and $\ell \leq m + O(1)$.*

Proof. A two-part description for x is almost a program for x. The first part is given by S^*, the second part is i_x, and the index of x in S. To make a well-defined program that computes x from the concatenation S^*i_x, one needs only a $O(1)$ instruction as a preamble. Note that the second part of the code does not need any additional prefix for self-delimitation, since its length $|i_S^x| = \lceil \log |S| \rceil$ can be computed (by the preamble) from S^*. The resulting program has a length of $m + O(1)$.

A bound on the running time is obtained on the two parts of the code. The first part runs for time $\leq B(\alpha)$, the most conservative bound for an α-bit program. The second part is fast: its running time is linear in $|S|$, so for models that are more elaborated than S_{Babel} (i.e., $\alpha \geq K(n)$) the running time is at most exponential in n, so smaller than $B(\alpha)$. The conclusion follows from $2B(\alpha) \leq B(\alpha + O(1))$. $\square$

Proposition A2. $\exists a \ (\tau, \ell)$-*program* $\implies \exists a \ (\alpha, m)$-*two part description, with $\alpha \leq \tau + K(\ell|\tau) + O(1)$ and $m \leq \ell + K(\tau|\ell) + O(1)$.*

Proof. From a (τ, ℓ)-program p for x, we can define the set of programs

$$M' = \{q : |q| = \ell \ \& \ B(\tau - 1) < RT(q) \leq B(\tau)\} \ni p,$$

from which we define the model $M = \{\mathcal{U}(q) : q \in M'\} \ni x$ serving as the (α, m)-two-part description.

We first bound $\alpha = K(M)$. Recalling Definition A2 in which $\bar{q}_\tau$ denotes the last halting program of length τ,

$$\alpha = K(M) \le K(M') \le K(\ell, \bar{q}_\tau).$$

This last expression will prove useful later, but it can be further upper-bounded by using the fact that $K(\bar{q}_\tau) = \tau + O(1)$. Indeed, it should be first observed that $K(\bar{q}_\tau) \ge \tau + O(1)$, since otherwise, executing the execution of $\bar{q}_\tau^*$ would be too short a way of computing for as long as $B(\tau)$. Second, $K(\bar{q}_\tau) \le |q_\tau| + O(1) = \tau + O(1)$, a statement which is stronger than the upper bound for prefix complexity given in Equation (A1). This stronger upper bound holds because $\bar{q}_\tau$ is not just any string, but a self-delimiting program, so $\bar{q}_\tau$ already contains within itself the information about its length. More precisely, the reference universal machine $\mathcal{U}$ could be transformed into an identity machine $\mathcal{V}$ defined on the same domain. Since $\mathcal{U}$ can then simulate $\mathcal{V}$, the conclusion follows. Therefore,

$$\alpha \le K(\bar{q}_\tau) + K(\ell | K(\bar{q}_\tau)) \le \tau + K(\ell | \tau).$$

Second, we bound $m = K(M) + \log |M|$. Consider any program $q \in M'$. Using the chain rule in two different ways,

$$\begin{aligned} K(q, \bar{q}_\tau, \ell) &= K(\bar{q}_\tau, \ell) + K(q | \bar{q}_\tau, \ell, K(\bar{q}_\tau, \ell)) \\ &= K(q) + K(\bar{q}_\tau, \ell | q, K(q)), \end{aligned}$$

one finds that

$$\begin{aligned} K(q | \bar{q}_\tau, \ell, K(\bar{q}_\tau, \ell)) &= K(q) + K(\bar{q}_\tau, \ell | q, K(q)) - K(\bar{q}_\tau, \ell) \\ &\le \ell + K(\tau | \ell) - K(\bar{q}_\tau, \ell). \end{aligned}$$

(Note that the chain rule as written in Equation (1) conditions on x^*. Here, we condition on $(x, K(x))$, which is informationally equivalent to x^*. In fact, $(x, K(x))$ is computed from x^* and a $O(1)$ advice which measures x^* before executing it, while x^* can be computed from $(x, K(x))$ by running in parallel all programs of length $K(x)$ until one of them produces x. This program is x^*.) The last line was obtained by noticing again that since q is a self-delimiting program, $K(q) \le |q| + O(1) = \ell + O(1)$ and by observing that $\bar{q}_\tau$ can be computed from q, τ and an $O(1)$ advice as there is only a constant number of τ-bit programs that halt after q—and $\bar{q}_\tau$ is the last one of them. See Proposition A4 for a proof of this claim.

In general, the number of strings s with $K(s | z) \le b$ is smaller than $2^{b+1} - 1$, because there are only this many programs that are short enough. Hence,

$$\log |M| \le \log |M'| \le \ell - K(\bar{q}_\tau, \ell) + K(\tau | \ell),$$

and so

$$m = K(M) + \log |M| \le \ell + K(\tau | \ell).$$

□

Proposition A2 is about the existence of a model whose (α, m) "coordinates" are not too far on the up-right of the (τ, ℓ) "coordinates" of a program. The aim, however, is to bound the region for the path of the time profile, given the description profile. Given a (α, m)-two-part description that is optimal (i.e., of minimal m for a given α), the time profile cannot admit (τ, ℓ)-programs with (τ, ℓ) too far on the lower-left of (α, m); otherwise, the aforementioned proposition would contradict the optimality of the (α, m)-two-part description. We know that "lower" is quantified by $K(\tau | \ell)$, while "left" by $K(\ell | \tau)$, but these quantities are expressed in terms of τ and ℓ, and need to be instead bounded in terms of α and m. This is what the following proposition does.

Proposition A3. $\forall \tau, \ell \in [0, n], \forall \alpha \in [\tau - O(1), \tau + K(\ell|\tau)]$ *and* $\forall m \in [\ell - O(1), \ell + K(\tau|\ell)],$

$$
\begin{aligned}
K(\ell|\tau) &\leq K(\alpha|m) + 2\ell^*(\ell^*(n)) \\
K(\tau|\ell) &\leq K(m|\alpha) + 2\ell^*(\ell^*(n)) \,.
\end{aligned}
$$

Proof. First,

$$
|\alpha - \tau| \leq K(\ell|\tau) \leq K(\ell) \leq \ell^*(n) \qquad \text{and} \qquad |m - \ell| \leq K(\tau|\ell) \leq K(\tau) \leq \ell^*(n) \,.
$$

Then,

$$
\begin{aligned}
K(\ell|\tau) &\leq K(m, |m - \ell| \,|\, \tau) \\
&\leq K(m|\tau) + K(|m - \ell|) \\
&\leq K(m|\alpha) + K(|\alpha - \tau|) + K(|m - \ell|) \\
&\leq K(m|\alpha) + 2\ell^*(\ell^*(n))
\end{aligned}
$$

$$
\begin{aligned}
K(\tau|\ell) &\leq K(\alpha, |\tau - \ell| \,|\, \ell) \\
&\leq K(\alpha|\ell) + K(|\alpha - \tau|) \\
&\leq K(\alpha|m) + K(|m - \ell|) + K(|\alpha - \tau|) \\
&\leq K(\alpha|m) + 2\ell^*(\ell^*(n)) \,.
\end{aligned}
$$

We define the quantities $\delta(\alpha)$ and $\epsilon(\alpha)$ as

$$
\begin{aligned}
\delta(\alpha) &:= K(\alpha | \alpha + h_x(\alpha)) + 2\ell^*(\ell^*(n)) + O(1), \\
\epsilon(\alpha) &:= K(\alpha + h_x(\alpha) | \alpha) + 2\ell^*(\ell^*(n)) + O(1) \,.
\end{aligned}
$$

For each point (α, m) of the boundary of the description profile can be drawn the course of $(\alpha + O(1), m + O(1))$, as well as $(\alpha - \delta(\alpha), m - \epsilon(\alpha))$. Note that a "drop" of the structure function is defined in Equation (6) precisely to ensure that the time profile has dropped of at least one. This ensures that the data x is logically deeper than expected from the minimal partial models of smaller complexity. The data thus contains more halting information that will be encompassed in the new minimal partial model. Indeed, as can be seen from Figure A1, there is a drop when the description profile drops by more than $Q(\alpha_{i-1}) = \epsilon(\alpha_{i-1}) + O(1)$, compared to the previous minimal partial model of complexity level α_{i-1}.
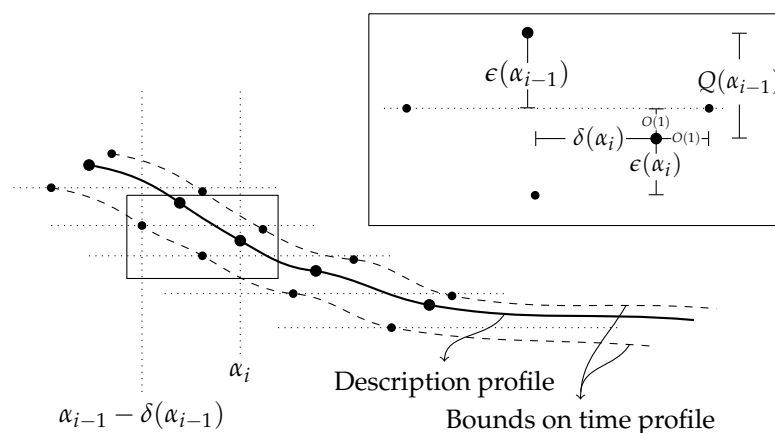

Figure A1. A visual help for the proof of Theorem 3.

More importantly for this proof, and also readable from Figure A1, the program S_i^* that computes the minimal partial model at complexity level α_i corresponding to the drop has a running time of at least $B(\alpha_{i-1} - \delta(\alpha_{i-1}) + O(1))$, since otherwise, the program

(determined by the two-part description) of length $m + O(1)$ runs in a time that contradicts the bounds for the time profile.

Hence, for $i < j$, S_i contains at least $\xi := \alpha_{i-1} - \delta(\alpha_{i-1}) + O(1)$ bits of irreducible halting information; namely, if ξ is given, the running time of S_i^* can be used to compute $\bar{q}_\xi$. Therefore,

$$\alpha_i + K(\xi|S_i^*) = K(S_i) + K(\xi|S_i^*) = K(S_i, \bar{q}_\xi) = K(\bar{q}_\xi) + K(S_i|\bar{q}_\xi, \xi)$$
$$= \alpha_{i-1} - \delta(\alpha_{i-1}) + K(S_i|\bar{q}_\xi, \xi),$$

and so

$$K(S_i|\bar{q}_\xi, \xi) = \alpha_i - \alpha_{i-1} + K(\xi|S_i^*) + \delta(\alpha_{i-1}) \tag{A3}$$

The model S_j contains more halting information than S_i, so if ξ is given, S_j^* can also be used to compute $\bar{q}_\xi$. Using Equation (A3),

$$
\begin{aligned}
K(S_i|S_j^*) &\leq K(\xi|S_j^*) + K(S_i|\bar{q}_\xi, \xi) \\
&= K(\xi|S_j^*) + K(\xi|S_i^*) + \alpha_i - \alpha_{i-1} + \delta(\alpha_{i-1}) \\
&\leq \alpha_i - \alpha_{i-1} + \delta(\alpha_{i-1}) + 2K(\xi) \\
&= \alpha_i - \alpha_{i-1} + K(\alpha_{i-1}) + 2K(\xi) + 2\ell^*\ell^*(n).
\end{aligned}
$$

Since

$$
\begin{aligned}
K(\xi) &\leq K(\alpha_{i-1}, K(\alpha_{i-1}), 2\ell^*\ell^*(n)) \\
&\leq K(\alpha_{i-1}) + K(K(\alpha_{i-1})|\alpha_{i-1}, K(\alpha_{i-1})) + K(\ell^*\ell^*(n))) \\
&\leq K(\alpha_{i-1}) + \ell^*\ell^*\ell^*(n),
\end{aligned}
$$

we find

$$
\begin{aligned}
K(S_i|S_j^*) &\leq \alpha_i - \alpha_{i-1} + 3K(\alpha_{i-1}) + 2(\ell^*\ell^*(n) + \ell^*\ell^*\ell^*(n)) \\
&\leq \alpha_i - \alpha_{i-1} + 3K(\alpha_{i-1}) + O(\log\log(n)).
\end{aligned}
$$

□

Lemma A2. *If $h_x(\alpha_i - 1) - h_x(\alpha_i) \geq Q(\alpha_i - 1)$, then the minimal partial model S_i that witnesses $h_x(\alpha_i)$ is such that for any $j > i$, $K(S_i|S_j) \leq 3K(\alpha_i) + O(\log\log n) \leq O(\log n)$.*

Proof. The proof is analogous to that of Theorem A3. The main difference resides in the bounds of the running time of S_i^* when the structure function follows a drastic drop, which in this case can be lower bounded by $B(\alpha_i - \delta(\alpha_i))$. See Figure A2 in contrast to Figure A1. □
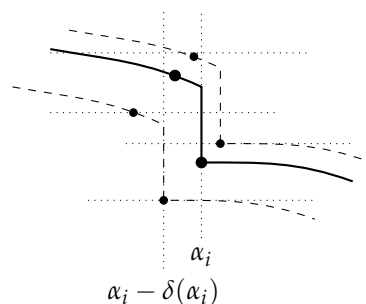

Figure A2. A visual help for the proof of Lemma A2.

Proposition A4.
$$|\{p\colon |p| = \tau \,\&\, RT(p) > B(\tau - 1)\}| = O(1)$$

Proof. The proof is by contradiction, exhibiting a too short program for $\bar{q}_{\tau-1}$. Suppose that the number of programs of length τ that halt after $\bar{q}_{\tau-1}$ is between 2^ν and $2^{\nu+1}$. Then, $\bar{q}_{\tau-1}$ can be obtained by running all programs of length τ and grouping them as they halt in groups of 2^ν. By assumption, there will be a full group completed with its last halting program running for longer than $\bar{q}_{\tau-1}$. If r denotes the number of this group, then $\bar{q}_{\tau-1}$ can be computed from τ, ν and r.

The map
$$\mu\colon\ \tau \mapsto 2^{-\tau} |\{p\colon |p| = \tau \,\&\, \mathcal{U}(p) \searrow\}|$$

is lower semi-computable and it is a semi-measure, since $\sum_\tau \mu(\tau) \le 1$. Thus, by the coding theorem [11], $\mu(\tau) \le 2^{-K(\tau)+O(1)}$. Hence,

$$r \le \frac{|\{p\colon |p| = \tau \,\&\, \mathcal{U}(p)\searrow\}|}{2^\nu} \le \frac{2^{\tau - K(\tau)+O(1)}}{2^\nu},$$

and so $|r| \le \tau - K(\tau) - \nu + O(1)$. Therefore, the following relations hold up to $O(1)$:

$$
\begin{aligned}
\tau - 1 = K(\bar{q}_{\tau-1}) &\le& K(\tau, \nu, r)\\
&\le& K(\tau, \nu) + K(|r| \,||\, \tau, \nu, K(\tau)) + K(r\,||\,r)\\
&\le& K(\tau, \nu) + K(r\,||\,r)\\
&\le& K(\tau, \nu) + |r|\\
&\le& K(\tau) + K(\nu | \tau, K(\tau)) + \tau - K(\tau) - \nu\\
&\le& K(\nu) + \tau - \nu.
\end{aligned}
$$

The above holds only if $\nu - K(\nu) = O(1)$, namely, only if $\nu = O(1)$. □

References

1. Lewes, G.H. *Problems of Life and Mind*; Facsimile Publisher: London, UK, 1875.
2. O'Connor, T.; Wong, H.Y. *Emergent Properties*; Zalta, E.N., Ed.; Stanford University: Stanford, CA, USA, 2012.
3. Bedau, M.A.; Humphreys, P.E. *Emergence: Contemporary Readings in Philosophy and Science*; MIT Press: Cambridge, MA, USA, 2008.
4. Available online: https://en.wikipedia.org/wiki/Emergence (accessed on 1 September 2021).
5. Ross, W.D. *Aristotle's Metaphysics. A Revised Text with Introduction and Commentary*; John Murray: London, UK, 1925.
6. Anderson, P.W. More is different. *Science* **1972**, *177*, 393–396. [CrossRef] [PubMed]
7. Wallace, D. *The Emergent Multiverse: Quantum Theory According to the Everett Interpretation*; Oxford University Press: Oxford, UK, 2012.
8. Dennett, D.C. Real patterns. *J. Philos.* **1991**, *88*, 27–51. [CrossRef]
9. Popper, K. *The Logic of Scientific Discovery*; Routledge: Thames, UK, 2005.
10. Gell-Mann, M.; Lloyd, S. Information measures, effective complexity, and total information. *Complexity* **1996**, *2*, 44–52. [CrossRef]
11. Li, M.; Vitányi, P. *An Introduction to Kolmogorov Complexity and Its Applications*; Springer: New York, NY, USA, 2008. [CrossRef]
12. Chaitin, G.J. The halting probability omega: Irreducible complexity in pure mathematics. *Milan J. Math.* **2007**, *75*, 291–304. [CrossRef]
13. Feynman, R.P. Simulating physics with computers. *Int. J. Theor. Phys.* **1982**, *21*, 467–488. [CrossRef]
14. Solomonoff, R.J. A formal theory of inductive inference. Part I. *Inf. Control* **1964**, *7*, 1–22. [CrossRef]
15. Kolmogorov, A.N. Three approaches to the quantitative definition of information. *Probl. Peredachi Inf.* **1965**, *1*, 3–11. [CrossRef]
16. Chaitin, G.J. On the length of programs for computing finite binary sequences. *J. ACM* **1966**, *13*, 547–569. [CrossRef]
17. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
18. Turing, A.M. On computable numbers, with an application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.* **1937**, *2*, 230–265. [CrossRef]
19. Kolmogorov, A.N. Talk at the Information Theory Symposium in Tallinn. Estonia (then USSR). 1974.
20. Vereshchagin, N.; Shen, A. Algorithmic statistics: Forty years later. In *Computability and Complexity*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 669–737.
21. Fisher, R.A. On the mathematical foundations of theoretical statistics. *Philos. Trans. R. Soc. A* **1922**, *222*, 309–368.
22. Antunes, L.; Bauwens, B.; Souto, A.; Teixeira, A. Sophistication vs. Logical Depth. *Theory Comput. Syst.* **2017**, *60*, 280–298. [CrossRef]

23. Koppel, M. Complexity, depth, and sophistication. *Complex Syst.* **1987**, *1*, 1087–1091.
24. Vitányi, P.M. Meaningful information. *IEEE Trans. Inf. Theory* **2006**, *52*, 4617–4626. [CrossRef]
25. Bennett, C.H. The thermodynamics of computation—A review. *Int. J. Theor. Phys.* **1982**, *21*, 905–940. [CrossRef]
26. Zurek, W.H. Algorithmic randomness and physical entropy. *Phys. Rev. A* **1989**, *40*, 4731. [CrossRef] [PubMed]
27. Maxwell, J.C. *Theory of Heat*; Cambridge University Press: Cambridge, UK, 1871.
28. Jaynes, E.T. The gibbs paradox. In *Maximum Entropy and Bayesian Methods*; Springer: Berlin/Heidelberg, Germany, 1992; pp. 1–21.
29. Rosenkrantz, R.D. *ET Jaynes: Papers on Probability, Statistics and Statistical Physics*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 158.
30. Baumeler, Ä.; Wolf, S. Causality–Complexity–Consistency: Can space-time be based on logic and computation? In *Time in Physics*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 69–101.
31. Ay, N.; Muller, M.; Szkola, A. Effective complexity and its relation to logical depth. *IEEE Trans. Inf. Theory* **2010**, *56*, 4593–4607. [CrossRef]
32. Vereshchagin, N.; Vitányi, P. Kolmogorov's structure functions with an application to the foundations of model selection. In Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, Vancouver, BC, Canada, 19 November 2002; pp. 751–760.
33. Gács, P.; Tromp, J.T.; Vitányi, P.M. Algorithmic statistics. *IEEE Trans. Inf. Theory* **2001**, *47*, 2443–2463. [CrossRef]
34. Callen, H.B.; Welton, T.A. Irreversibility and generalized noise. *Phys. Rev.* **1951**, *83*, 34. [CrossRef]
35. Caux, J.S.; Mossel, J. Remarks on the notion of quantum integrability. *J. Stat. Mech. Theory Exp.* **2011**, *2011*, P02023. [CrossRef]
36. Chaitin, G.J. Information-theoretic incompleteness. In *Applied Mathematics and Computation*; Elsevier: Amsterdam, The Netherlands, 1992; Volume 52, pp. 83–101.
37. Chaitin, G.J. *Meta Maths! The Quest for Omega*; Pantheon Books: New York, NY, USA, 2005.
38. Shen, A.; Uspensky, V.A.; Vereshchagin, N. *Kolmogorov Complexity and Algorithmic Randomness*; MCCME: Moscow, Russia, 2013.
39. Bennett, C.H. Logical depth and physical complexity. In *The Universal Turing Machine: A Half-Century Survey*; Oxford University Press: New York, NY, USA, 1988; pp. 227–257.