What Is Computable? What Is Feasibly Computable? A Physicist's Viewpoint

Vladik Kreinovich and Olga Kosheleva

Abstract In this chapter, we show how the questions of what is computable and what is feasibly computable can be viewed from the viewpoint of physics: what is computable within the current physics? what is computable if we assume – as many physicists do – that no final physical theory is possible? what is computable if we consider data processing, i.e., computations based on physical inputs? Our physics-based analysis of these questions leads to some unexpected answers, both positive and negative. For example, we show that under the no-physical-theory-is-perfect assumption, almost all problems are feasibly solvable – but not all of them.

1 What Is Computable? What Is Feasibly Computable? Different Aspects of These Questions

The two main questions of theoretical computer science. One of the main objectives of theoretical computer science is to answer the following two fundamental questions:

- The first question is: which tasks are computable in principle?
- Once we learned that a task is, in principle, computable, a natural next question is: is this task *feasibly* computable, i.e., can we perform the corresponding computations in reasonable time?

These questions are usually considered from the viewpoint of a computer scientist. From the viewpoint of a computer scientist, computation is a solution to a well-defined task, performed on a well-defined computational devices. As a result, when formulating and analyzing the above problems, computer scientists usually

Vladik Kreinovich and Olga Kosheleva
University of Texas at El Paso, 500 W. University, El Paso, Texas 79968, USA
e-mails: vladik@utep.edu, olgak@utep.edu

consider well-defined tasks and computations which consist of a sequence of well-defined elementary steps.

A physicist's understanding is somewhat different. Computer science is, after all, an applied discipline. From the practical viewpoint, we need computations to process data from the real world – so that we will be able to predict the future state of the world and, in situations when we can control this future state, to come up with actions that would result in the best possible outcome.

From this viewpoint, we can distinguish between two different types of computations:

- traditional computations, when we are trying to find a solution to a well-defined (= mathematical) problem, and
- data processing computations, when we process the data coming from the physical world.

Similarly, based on what computational devices we can use, we can distinguish between two possible approaches:

- a "purist" approach, when we are only allowed step-by-step computations on a well-defined computational device, and
- a pragmatic approach, when, in addition to computations, we can set up physical models of the analyzed systems, analog computations – whatever helps.

Thus, each of the two fundamental questions – what is computable? what is feasibly computable? – can be formulated in three different ways. Namely, in addition to the traditional formulation, when we consider computing well-defined mathematical tasks on well-defined computers, we can also consider:

- a pragmatic formulation, when, in addition to well-defined computers, we can use physical processes to help with computations, and, finally
- a data processing formulation when we are interested in processing physical data.

What we do in this chapter. In this chapter, we consider all these three approaches one by one, and we show that they lead to somewhat different answers to the fundamental questions of what is computable and what is feasible computable.

The structure of this chapter is as follow. The pragmatic formulation is discussed in Sections 2 and 3, and the data processing formulation is discussed in Section 4.

2 Non-Standard Physical Processes Can Help Computations: Examples Based on Specific Physical Models

Solving NP-complete problems is important. In practice, we often need to find a solution that satisfies a given set of constraints – or at least check that such a solution is possible. Once we have a candidate for the solution, we can feasibly check whether this candidate indeed satisfies all the constraints. In theoretical computer

science, "feasibly" is usually interpreted as computable in polynomial time, i.e., in time bounded by a polynomial of the length of the input.

A problem of checking whether a given set of constraints has a solution is called a problem of the class NP if we can check, in polynomial time, whether a given candidate is a solution; see, e.g., [26].

Examples of such problems includes checking whether a given graph can be colored in 3 colors, checking whether a given propositional formula – i.e., formula of the type

$$(v_1 \vee \neg v_2 \vee v_3) \,\&\, (v_4 \vee \neg v_2 \vee \neg v_5) \,\&\, \ldots,$$

is satisfiable, i.e., whether this formula is true by some combination of the propositional variables v_i, etc.

Each problem from the class NP can be algorithmically solved by trying all possible candidates. For example, we can check whether a graph can be colored by trying all possible assignments of colors to different vertices of a graph, and we can check whether a given propositional formula is satisfiable by trying all 2^n possible combinations of true-or-false values $v_1, \ldots, v_n$. Such exhaustive search algorithms require computation time like 2^n, time that grows exponentially with n. For medium-size inputs, e.g., for $n \approx 300$, the resulting time is larger than the lifetime of the Universe. So, these exhaustive search algorithms are not practically feasible.

It is not known whether problems from the class NP can be solved feasibly (i.e., in polynomial time): this is a famous open problem $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$. It is known, however, that there are problems in the class NP which are *NP-complete* in the sense that every problem from the class NP can be reduced to this problem. Reduction means, in particular, that if we can find a way to efficiently solve one NP-complete problem, then, by reducing other problems from the class NP to this problem, we can thus efficiently solve all the problems from the class NP.

So, it is very important to be able to efficiently solve even one NP-complete problem. (By the way, both above example of NP problems – checking whether a graph can be colored in 3 colors and whether a propositional formula is satisfiable – are NP-complete.)

Can the use of non-standard physics speed up the solution of NP-complete problems? NP-completeness of a problem means, crudely speaking, that the problem may take an unrealistically long time to solve – at least on computers based on the usual physical techniques. A natural question is: can the use of non-standard physics speed up the solution of these problems?

To answer this question, let us start the analysis of the corresponding physics.

Parallelization: a natural idea. If a person faces a task that would take too much time for him or her working alone – e.g., building a house – this person asks for help. Similarly, when a problem takes too much time to solve on a single computer, a natural idea is to have several computers working on this problem in parallel.

Physical limitations to parallelization speed-up. At first glance, potentially, by dividing the original problem into smaller and smaller pieces and using more and

more processors to process these pieces, we can speed up the computation as much as possible.

In reality, however, there are physical limitations on the possible speed-up; see, e.g., [24]. Indeed, let us assume that we have a parallel algorithm that, for all inputs of bit length $\le n$, solves the original problem in time $T_{\rm par}(n)$.

The user is located at some point in space. The user inputs the problem at this spatial location, and the user expects the result of the computation to be delivered to the same spatial location. Each processor that participates in the desired computation must:

- get this signal (directly or indirectly) from the user's location, and then
- start some other signals that will eventually reach the user at his or her spatial location.

So, if a processor is located at distance r from the user, then the signal going from the user to the processor and back must cover the distance of at least $2r$.

According to modern physics, the speed of all communications is limited by the speed of light c. Thus, the smallest amount of time for this signal transmission is $\dfrac{2r}{c}$. If this time exceeds $T_{\rm par}(n)$, this means that this processor is unable to contribute to the computation result. Thus, only processors for which $\dfrac{2r}{c} \le T_{\rm par}(n)$, i.e., for which $r \le R(n) \stackrel{\rm def}{=} \dfrac{1}{2}\cdot c \cdot T_{\rm par}(n)$, contribute to the computation. So, we only need to consider processors which are located inside the sphere of radius $R(n)$ centered at the user.

How many processors can fit inside this sphere? A physical bound of the number $N_{\rm proc}(n)$ of these processors can be obtained if we divide the volume $V(R(n))$ of the inside of this sphere by the smallest possible volume ΔV of a processor:

$$N_{\rm proc} \le \frac{V(R(n))}{\Delta V}.$$

In the Euclidean space, $V(R) = \dfrac{4}{3}\cdot\pi\cdot R^3$, so we conclude that

$$N_{\rm proc}(n) \le \frac{1}{\Delta V}\cdot\frac{4}{3}\cdot\pi\cdot(R(n))^3 = \frac{1}{\Delta V}\cdot\frac{4}{3}\cdot\pi\cdot\frac{1}{8}\cdot c^3\cdot(T_{\rm par}(n))^3,$$

i.e., that $N_{\rm proc}(n) \le {\rm const}\cdot(T_{\rm par}(n))^3$, where the multiplicative constant does not depend on the size n of the input.

We can always simulate parallel computations by $N_{\rm par}(n)$ processors on a sequential machine: for this, for each original cycle of the parallel machine, we need to emulate how the state of each of $N_{\rm proc}(n)$ processors change. In this simulation, one step of the original parallel machine requires $N_{\rm proc}(n)$ steps of the simulating sequential machine. Thus, the overall time $T_{\rm seq}(n)$ of the corresponding sequential machine can be obtained by multiplying the original parallel time $T_{\rm par}(n)$ by the number of processors: $T_{\rm seq}(n) \le T_{\rm par}(n)\cdot N_{\rm proc}(n)$. By using the bound $N_{\rm proc}(n) \le {\rm const}\cdot(T_{\rm par}(n))^3$, we conclude that

$$T_{\text{seq}}(n) \le \text{const} \cdot (T_{\text{par}}(n))^4.$$

So, if a problem is difficult to solve on a sequential machine, and there is a huge lower bound on $T_{\text{seq}}(n)$, then we can conclude that there is a related lower bound on the parallel time as well: $T_{\text{par}}(n) \ge \text{const} \cdot (T_{\text{seq}}(n))^{1/4}$. In particular, if – as most computer scientists believe – an NP-complete problem cannot be solved faster than in exponential time $T_{\text{seq}}(n) \ge 2^n$, then we get similar exponential lower bounds on the parallel time as well: $T_{\text{par}}(n) \ge (\sqrt[4]{2})^n$. This is faster than 2^n, but still not feasible.

Important observation: these limitations depend on physics. The above limitations are based on the usual physics, where the space is Euclidean (so that the volume grows as a cube of the radius), and the speeds of all physical processors are limited by the speed of light.

However, it is well known that the actual space-time is different from Euclidean, it is *curved*; see, e.g., [5]. Also, physicists are seriously considering space-time models in which it is possible to exceed the speed of light; see, e.g., [29]. This leads to the possibility of potentially physically realistic situations in which we can solve NP-complete problems in polynomial time. Let us briefly enumerate such situations.

Case of curved space-time. Already in the historically very first non-Euclidean geometry – the hyperbolic Lobachevsky space – the volume $V(R)$ of the inside of the sphere grows exponentially with radius. Thus, in principle, we can fit exponentially many processors within a radius that grows linearly with n. On the resulting parallel machine, if we ask each processor to check one of 2^n Boolean vectors, we can thus solve the NP-complete propositional satisfiability problem in linear time; see, e.g., [21, 24]. So, if the proper physical space is hyperbolic, we can solve NP-complete problems in polynomial time.

Another possible scheme is related to the "almost" black holes [24]. One of the well-known consequences of general relativity is the existence of the "black hole" solutions. A black hole is an area from which nothing comes out (in particular, light cannot escape it, hence it looks black). It is proved that if an object (e.g., a star) is massive enough, it will eventually be crushed by its own gravitational force and form a black hole. If the object is smaller, or if it has a significant electric charge, then it forms an "almost" black hole, i.e., an area from which it is possible but difficult to escape. If you enter this "almost" black hole, you go into the narrow throat; see, e.g., [23], Chapters 31 and 44. From the outside, it looks like a small particle. So, a natural hypothesis (described in Chapter 44 of [23]) is that all charged elementary particles *are* actually such "almost" black holes.

Each of these throats is gateway to a different space-time. So, to solve a propositional satisfiability problem with n variables $v_1, \ldots, v_n$, we can pick up two particles in our world – which are gateways to different worlds – and:

- ask the folks from the first of these worlds to check the propositional satisfiability of a formula obtained when we plug in $v_n =$"true", and
- ask the second world to do the same with $v_n =$"false".

Participants living in each of these world will thus be given a formula with $n-1$ Boolean variables. To check the satisfiability of each of these formulas, they will repeat the same procedure: find two particles-gateways in their world, and ask the corresponding creatures to solve a problem with $n-2$ variables, etc. In n steps, we reduce the problem to checking a formula with a single variable, and we need n steps to send the results back. Thus, in linear time, we also get a solution to the propositional satisfiability problem (because, as one can see, in this world with almost black holes, the volume $V(R)$ also grows exponentially with the radius R).

Possibility of velocities exceeding speed of light. If we allow processes exceeding speed of light, then we have acausal processes, i.e., the possibility to go back to the past [29]. The simplest thing that we can do in this case is to let a slow computer solve the problem for as long as it takes – and then send the result back in time, so that the user will get it right after he or she requested the solution. More sophisticated schemes are also possible; see, e.g., [10, 11].

Other possible schemes of using non-standard physics to speed up computations. To speed up computations, we can also use the fact that, according to relativity theory, time slows down when one travels at a speed close to speed of light or in a strong gravitational field (e.g., near the black hole). So, if the whole civilization starts going around at a speed close to the speed of light and/or moves close to the black hole, then, by performing computations on stationary planets far away from the black hole, we get the result much faster – in terms of our time. For example, if 1 year for us will be 10 years for the outside world, then a problem that takes 10 years to compute will be solved after 1 year of our time.

Other possible schemes include the use of quantum effects, etc.; see, e.g., [1, 28].

3 What If No Final Theory Is Possible?

In the previous section, we analyzed how specific physical phenomena affect computability. In this analysis, we considered several specific physical models, such as cosmological solutions with wormholes and/or casual anomalies, etc. However, many physicists believe that no physical theory is perfect, i.e., that no matter how many observations support a physical theory, inevitably, new observations will come which will require this theory to be updated. In this section, following [13, 14, 20, 31], we prove that if such a no-perfect-theory principle is true, then the use of physical data

- can enhance computations, and
- can drastically speed up the solution of NP-complete problems: namely, we can feasibly solve almost all instances of each NP-complete problem.

3.1 No Physical Theory Is Perfect: How to Formalize the Widely Spread Physicists' Belief

No physical theory is perfect: a widely spread physicists' belief. If we prove that, within a given physical theory, we can speed up the solution to NP-complete problems, will this answer be fully satisfactory?

So far, in the history of physics, no matter how good a physical theory, no matter how good its accordance with observations, eventually, new observations appear which are not fully consistent with the original theory – and thus, a theory needs to be modified. For example, for several centuries, Newtonian physics seems to explain all observable facts – until later, quantum (and then relativistic) effects were discovered which required changes in physical theories.

Because of this history, many physicists believe that every physical theory is approximate – no matter how sophisticated a theory, no matter how accurate its current predictions, inevitably new observations will surface which would require a modification of this theory; see, e.g., [5].

How does this belief affect computations? At first glance, the fact that no theory is perfect makes the question of possible computation of non-computable sequences and of possible speed-up rather hopeless: no matter how good results we achieve within a given physical theory, eventually, this theory will turn out to be, strictly speaking, false – and thus, our computation or speed-up schemes will not be applicable.

In this section, we show, however, that in spite of this seeming hopelessness, an important non-standard computations and speed-up results can be deduced simply from the fact no physical theory is perfect.

How to describe, in precise terms, that no physical theory is perfect: discussion. The statement that no physical theory is perfect means that no matter what physical theory we have, eventually there will be observations which violate this theory. To formalize this statement, we need to formalize what are observations and what is a theory.

What are observations? Each observation can be represented, in the computer, as a sequence of 0s and 1s; actually, in many cases, the sensors already produce the signal in the computer-readable form, as a sequence of 0s and 1s.

An exact description of each experiment can also be described in precise terms, and thus, it will be represented in a computer as a sequence of 0s and 1s. An experiment should specify how long we wait for the result; in this way, we are guaranteed that we get the result.

In each experiment, we can specify which bit of the result we are interested in; for convenience, we can consider producing different bits as different experiments.

Each such experiment is represented as a sequence of 0s and 1s; by appending 1 at the beginning of this sequence, we can view this sequence as a binary expansion of a natural number i. This natural number will serve as the "code" describing the experiment. For example, a sequence 001 is transformed into $i = 1001_2 = 9_{10}$. (We

need to append 1, because otherwise two different sequences 001 and 01 will be represented by the same integer).

For natural numbers i which correspond to experiment descriptions, let ω_i denote the bit result of the experiment described by the code i.

Let us also define ω_i for natural numbers i which do not correspond to a syntactically correct description of experiments. For example, we can fix a scheme of an experiment that uses a natural number i as a parameter (e.g., repeating a certain procedure i times), and define ω_i as the result of this scheme.

In these terms, all past and future observations form a (potentially) infinite sequence $\omega = \omega_1\omega_2\ldots$ of 0s and 1s, $\omega_i \in \{0,1\}$.

Comment. To make sure that the resulting ph-algorithm is feasible, we need to define experiment descriptions in which a way that the time needed to complete the i-th experiment does not exceed a polynomial of $\log(i)$. From this viewpoint, an experiment in which there is no explicit time limit should be described as a sequence of experiments with a cutoff time $1, 2, \ldots, t, \ldots$; the allocated time can be indicated, e.g., by adding t special time symbols to the original description of the experiment.

What is a physical theory from the viewpoint of our problem: a set of sequences. A physical theory may be very complex, but all we care about is which sequences of observations ω are consistent with this theory and which are not. In other words, for our purposes, we can identify a physical theory T with the set of all sequences ω which are consistent with this theory.

Not every set of sequences corresponds to a physical theory: the set T must be non-empty and definable. Not every set of sequences comes from a physical theory. First, a physical theory must have at least one possible sequence of observations, i.e., the set T must be *non-empty*.

Second, a theory – and thus, the corresponding set – must be described by a finite sequence of symbols in an appropriate language. Sets which are uniquely by (finite) formulas are known as *definable*. Thus, the set T must be definable.

Since at any moment of time, we only have finitely many observations, the set T must be closed. Another property of a physical theory comes from the fact that at any given moment of time, we only have finitely many observations, i.e., we only observe finitely many bits. From this viewpoint, we say that observations $\omega_1\ldots\omega_n$ are consistent with the theory T if there is a continuing infinite sequence which is consistent with this theory, i.e., which belongs to the set T.

The only way to check whether an infinite sequence $\omega = \omega_1\omega_2\ldots$ is consistent with the theory is to check that for every n, the sequences $\omega_1\ldots\omega_n$ are consistent with the theory T. In other words, we require that for every infinite $\omega = \omega_1\omega_2\ldots$,

- if for every n, the sequence $\omega_1\ldots\omega_n$ is consistent with the theory T, i.e., if for every n, there exists a sequence $\omega^{(m)} \in T$ which has the same first n bits as ω, i.e., for which $\omega_i^{(m)} = \omega_i$ for all $i = 1,\ldots,n$,
- then the sequence ω itself should be consistent with the theory, i.e., this infinite sequence should also belong to the set T.

From the mathematical viewpoint, we can say that the sequences $\omega^{(m)}$ converge to ω: $\omega^{(m)} \to \omega$ (or, equivalently, $\lim \omega^{(m)} = \omega$), where convergence is understood in terms of the usual metric on the set of all infinite sequences $d(\omega, \omega') \stackrel{\text{def}}{=} 2^{-N(\omega,\omega')}$, where $N(\omega, \omega') \stackrel{\text{def}}{=} \max\{k : \omega_1 \ldots \omega_k = \omega'_1 \ldots \omega'_k\}$.

In general, if $\omega^{(m)} \to \omega$ in the sense of this metric, this means that for every n, there exists an integer ℓ such that for every $m \geq \ell$, we have $\omega_1^{(m)} \ldots \omega_n^{(m)} = \omega_1 \ldots \omega_n$. Thus, if $\omega^{(m)} \in T$ for all m, this means that for every n, a finite sequence $\omega_1 \ldots \omega_n$ can be a part of an infinite sequence which is consistent with the theory T. In view of the above, this means that $\omega \in T$.

In other words, if $\omega^{(m)} \to \omega$ and $\omega^{(m)} \in T$ for all m, then $\omega \in T$. So, the set T must contain all the limits of all its sequences. In topological terms, this means that the set T must be *closed*.

A physical theory must be different from a fact and hence, the set T must be nowhere dense. The assumption that we are trying to formalize is that no matter how many observations we have which confirm a theory, there eventually will be a new observation which is inconsistent with this theory. In other words, for every finite sequence $\omega_1 \ldots \omega_m$ which is consistent with the set T, there exists a continuation of this sequence which does not belong to T. The opposite would be if all the sequences which start with $\omega_1 \ldots \omega_m$ belong to T; in this case, the set T will be *dense* in the open set of all the sequences starting with $\omega_1 \ldots \omega_m$. Thus, in mathematical terms, the statement that every finite sequence which is consistent with T has a continuation which is not consistent with T means that the set T is *nowhere dense*.

Resulting definition of a theory. By combining the above properties of a set T which describes a physical theory, we arrive at the following definition.

Definition 3.1. *By a* physical theory*, we mean a non-empty closed nowhere dense definable set T.*

Mathematical comment. To properly define what is definable, we need to have a consistent formal definition of definability. In this paper, we follow a natural definition from [16, 15, 19, 20, 31] – which is reproduced in the Appendix.

Formalization of the principle that no physical theory is perfect. In terms of the above notations, the no-perfect-theory principle simply means that the infinite sequence ω (describing the actual results of all observations) is not consistent with any physical theory, i.e., that the sequence ω does not belong to any physical theory T. Thus, we arrive at the following definition.

Definition 3.2. *We say that an infinite binary sequence ω is* consistent with the no-perfect-theory principle *if the sequence ω does not belong to any physical theory (in the sense of Definition 3.1).*

Comment. Are there such sequences in the first place? Our answer is yes. Indeed, by definition, we want a sequence which does not belong to a union of all definable physical theories. Every physical theory is closed nowhere dense set. Every

definable set is defined by a finite sequence of symbols, so there are no more than countably many definable theories. Thus, the union of all definable physical theories is contained in a union of countably many closed nowhere dense sets. Such sets are knows as *meager* (or *Baire first category*); it is known that the set of all infinite binary sequences is not meager. Thus, there are sequences who do not belong to the above union – i.e., sequences which are consistent with the no-perfect-theory principle; see, e.g., [9, 25].

3.2 The Use of Physical Computations Can Enhance Computations

How to describe general computations. Each computation is a solution to a well-defined problem. As a result, each bit in the resulting answer satisfies a well-defined mathematical property. All mathematical properties can be described, e.g., in terms of Zermelo-Fraenkel theory ZF, the standard formalization of set theory. For each resulting bit, we can formulate a property P which is true if and only if this bit is equal to 1. In this sense, each bit in each computation result can be viewed as the truth value of some statement formulated in ZF. Thus, our general ability to compute can be described as the ability to (at least partially) compute the sequence of truth values of all statements from ZF.

All well-defined statements from ZF can be numbered, e.g., in lexicographic order. Let α_n denote the truth value of the n-th ZF statement, and let $\alpha = \alpha_1 \ldots \alpha_n \ldots$ denote the infinite sequence formed by these truth values. In terms of this sequence, our ability to compute is our ability to compute the sequence α.

Kolmogorov complexity as a way to describe what is easier to compute. We want to analyze whether the use of physical observations (i.e., of the sequence ω analyzed in the previous section) can simplify computations. A natural measure of easiness-to-compute was invented by A. N. Kolmogorov, the founder of modern probability theory, when he realized that in the traditional probability theory, there is no formal way to distinguish between:

- finite sequences which come from observing from truly random processes, and
- orderly sequences like $0101\ldots01$.

Kolmogorov noticed that an orderly sequence $0101\ldots01$ can be computed by a short program, while the only way to compute a truly random sequence $0101\ldots$ is to have a print statement that prints this sequence. He suggested to describe this differences by introducing what is now known as *Kolmogorov complexity* $K(x)$ of a finite sequence x: the shortest length of a program (in some programming language) which computes the sequence x.

- For an orderly sequence x, the Kolmogorov complexity $K(x)$ is much smaller than the length $\mathrm{len}(x)$ of this sequence: $K(x) \ll \mathrm{len}(x)$.
- For a truly random sequence x, we have $K(x) \approx \mathrm{len}(x)$; see, e.g., [22].

The smaller the difference $\mathrm{len}(x) - K(x)$, the more we are sure that the sequence x is truly random.

Relative Kolmogorov complexity as a way to describe when using an auxiliary sequence simplifies computations. The usual notion of Kolmogorov complexity provides the complexity of computing x "from scratch". A similar notion of the *relative Kolmogorov complexity* $K(x\,|\,y)$ can be used to describe the complexity of computing x when a (potentially infinite) sequence y is given. This relative complexity is based on programs which are allowed to use y as a subroutine, i.e., programs which, after generating an integer n, can get the n-th bit y_n of the sequence y by simply calling y. When we compute the length of such programs, we just count the length of the call, not the length of the auxiliary program which computes y_n – just like when we count the length of a C++ program, we do not count how many steps it takes to compute, e.g., $\sin(x)$, we just count the number of symbols in this function call. The relative Kolmogorov complexity is then defined as the shortest length of such a y-using program which computes x.

Clearly, if x and y are unrelated, having access to y does not help in computing x, so $K(x\,|\,y) \approx K(x)$. On the other hand, if x coincides with y, then the relative complexity $K(x\,|\,y)$ is very small: all we need is a simple for-loop, in which we call for each bit y_i, $i = 1, \ldots, n$, and print this bit right away.

Resulting reformulation of our question. In terms of relative Kolmogorov complexity, the question of whether observations enhance computations is translated into checking whether $K(\alpha_1 \ldots \alpha_n\,|\,\omega) \approx K(\alpha_1 \ldots \alpha_n)$ (in which case there is no enhancement) or whether $K(\alpha_1 \ldots \alpha_n\,|\,\omega) \ll K(\alpha_1 \ldots \alpha_n)$ (in which case there is a strong enhancement). The larger the difference $K(\alpha_1 \ldots \alpha_n) - K(\alpha_1 \ldots \alpha_n\,|\,\omega)$, the larger the enhancement.

Enhancement is possible. Let us show that under the no-perfect-theory principle, observations do indeed enhance computations.

Proposition 3.1. *Let α be a sequence of truth values of ZF statements, and let ω be an infinite binary sequence which is consistent with the no-perfect-theory principle. Then, for every integer $C > 0$, there exists an integer n for which $K(\alpha_1 \ldots \alpha_n\,|\,\omega) < K(\alpha_1 \ldots \alpha_n) - C$.*

Comment. In other words, in principle, we can have an arbitrary large enhancement.

3.3 The Use of Physical Observations Can Help in Solving NP-Complete Problems

Towards the main result of this section: that the use of physical observations can help in solving NP-complete problems. In this section, we prove that under the no-perfect-theory principle, it is possible to drastically speed up the solution of NP-complete problems.

How to represent instances of an NP-complete problem. For each NP-complete problem $\mathscr{P}$, its instances are sequences of symbols. In the computer, each such sequence is represented as a sequence of 0s and 1s. Thus, as in the previous section, we can append 1 in front of this sequence and interpret the resulting sequence as a binary code of a natural number i.

In principle, not all natural numbers i correspond to instances of a problem $\mathscr{P}$; we will denote the set of all natural numbers which correspond to such instances by $S_{\mathscr{P}}$.

For each $i \in S_{\mathscr{P}}$, the correct answer (true or false) to the i-th instance of the problem $\mathscr{P}$ will be denoted by $s_{\mathscr{P},i}$.

Easier-to-solve and harder-to-solve NP-complete problems. We will show that our method works on "harder-to-solve" NP-complete problems, harder-to-solve in the following sense. By definition, for all NP-complete problems, unless P = NP, there is no feasible algorithm for solving all its instances. However, for some easier-to-solve problems, there are feasible algorithms which solve "almost all" instances, in the sense that for each n, the proportional of instance $i \le n$ for which the problem is solved by this algorithm tends to 1. In this case, while the worst-case complexity is still exponential, in practice, almost all problems can be feasibly solved.

A more challenging case are harder-to-solve NP-complete problems, for which no feasible algorithm is known that would solve almost all instances.

In this section, we show that our method works on all NP-complete problems, both easier-to-solve and harder-to-solve ones.

What we mean by using physical observations in computations. In addition to performing computations, our computational device can produce a scheme i for an experiment, and then use the result ω_i of this experiment in future computations. In other words, given an integer i, we can produce ω_i.

In precise theory-of-computation terms, the use of physical observations in computations thus means computations that use the sequence ω as an *oracle*; see, e.g., [26].

Definition 3.3. *By a* ph-algorithm $\mathscr{A}$*, we mean an algorithm which uses, as an oracle, a sequence* ω *which is consistent with the no-perfect-theory principle.*

Notation. The result of applying an algorithm $\mathscr{A}$ using ω to an input i will be denoted by $\mathscr{A}(\omega, i)$.

Definition 3.4. *Let* $\mathscr{P}$ *be an NP-complete problem. We say that a feasible ph-algorithm* $\mathscr{A}$ solves almost all instances of $\mathscr{P}$ *if for every* $\varepsilon > 0$*, and for every natural number n, there exists an integer* $N \ge n$ *for which the proportion of the instances* $i \le N$ *of the problem* $\mathscr{P}$ *which are correctly solved by* $\mathscr{A}$ *is greater than* $1-\varepsilon$*:*

$$\forall \varepsilon > 0 \, \forall n \, \exists N \left(N \ge n \,\&\, \frac{\#\{i \le N : i \in S_{\mathscr{P}} \,\&\, \mathscr{A}(\omega, i) = s_{\mathscr{P},i}\}}{\#\{i \le N : i \in S_{\mathscr{P}}\}} > 1 - \varepsilon \right).$$

Comment. The restriction to sufficiently long inputs $N \geq n$ makes perfect sense: for short inputs, NP-completeness is not an issue: we can perform exhaustive search of all possible bit sequences of length 10, 20, and even 30. The challenge starts when the length of the input is high.

Proposition 3.2. *For every NP-complete problem $\mathscr{P}$, there exists a feasible ph-algorithm $\mathscr{A}$ that solves almost all instances of $\mathscr{P}$.*

Comments. In other words, we show that the use of physical observations makes all NP-complete problems easier-to-solve (in the above-described sense).

It turns out that this result is the best possible, in the sense that the use of physical observations cannot solve *all* instances.

Proposition 3.3. *If P≠NP, then no feasible ph-algorithm $\mathscr{A}$ can solve all instances of $\mathscr{P}$.*

Comment. Another possible idea of strengthening Proposition 3.2 is to require that the property

$$\frac{\#\{i \leq N : i \in S_{\mathscr{P}} \,\&\, \mathscr{A}(\omega, i) = s_{\mathscr{P},i}\}}{\#\{i \leq N : i \in S_{\mathscr{P}}\}} > 1 - \varepsilon$$

hold not only for *infinitely many N*, but for *all N* starting with some N_0. It turns out that in this formulation, the use of physical observation does not help.

Definition 3.5. *Let $\mathscr{P}$ be an NP-complete problem. Let $\delta > 0$ be a real number. We say that a feasible ph-algorithm $\mathscr{A}$* δ-solves *$\mathscr{P}$ if*

$$\exists N_0 \, \forall N \left(N \geq N_0 \to \frac{\#\{i \leq N : i \in S_{\mathscr{P}} \,\&\, \mathscr{A}(\omega, i) = s_{\mathscr{P},i}\}}{\#\{i \leq N : i \in S_{\mathscr{P}}\}} > \delta \right).$$

Proposition 3.4. *For every NP-complete problem $\mathscr{P}$ and for every $\delta > 0$, if there exists a feasible ph-algorithm $\mathscr{A}$ that δ-solves $\mathscr{P}$, then there exists a feasible algorithm $\mathscr{A}'$ (not using physical observations) which also δ-solves $\mathscr{P}$.*

4 What If We Take into Account that We Are Only Interested in Processing Physical Data

Many physical theories accurately predict which events are possible and which are not, or – in situations where probabilistic (e.g., quantum) effects are important – predict the probabilities of different possible outcomes. At first glance, it may seem that this probabilistic information is all we need.

In this section, we show, however, that to adequately describe physicists' reasoning, it is important to also take into account additional physical knowledge – about what is possible and what is not. We show that, if we limit ourselves to objects which are physically possible, then many seemingly undecidable problems become algorithmically decidable.

How physicists make their conclusions: why probabilities are sometimes not enough. Modern physics makes many very accurate predictions of different events. In situations when quantum effects are important and thus, deterministic predictions are not possible, physics predicts probabilities of different events; see, e.g., [5]. There are still many problems where we cannot accurately predict the events and/or their probabilities, but in many other situations, the accuracy of predictions is truly amazing.

At first glance, once we know the probabilities, we are done: we can thus predict the frequencies with which the corresponding events will occur in real life. In many situations, probabilities are indeed all we need. For example, when we predict that the probability of a coin falling heads is 1/2, this means that in half of the cases, the coin will fall heads, in half, tails, and there is no other information that we can extract from observing the results of an actual coin toss: these results should be random.

This is true not only for coin tossing, but also for other predictions in which the predicted probability is "reasonable", i.e., not too small and not too close to 1. However, the situation is somewhat different when it comes to events with a very small probability. Let us give a few simple examples.

According to statistical physics, entropy of a closed system can only increase. This means, for example, that if we place a cold kettle on a cold stove, it is not possible that the kettle will start boiling by itself, while the stove will get colder – although this transfer of energy from the stove to the kettle does not contradict to the energy conservation law.

How do physicists conclude that this is not possible? They estimate the probability of such an event and conclude that this probability is extremely small. From the purely mathematical viewpoint, the fact that this probability is not zero means that if we wait long enough, then we will still see a kettle boiling on a cold stove. However, this is not what the physicists claim. What they claim is that the kettle *cannot* boil. In other words, they claim that the corresponding event is simply not possible [3, 4, 5].

Another example is the impossibility of spontaneous human levitation. The fact that a body has a non-zero temperature means that all the atoms and all the molecules in the body are randomly oscillating. Again, since all the molecules are going in random directions, there is a non-zero probability that they will all go into the same direction and a person will be spontaneously lifted above ground. What the physicists claim is not that such a possibility is rare, they claim is that it is simply not possible.

Physicists make similar conclusions about all irreversible events. For example, if we place a gas in one half of the box and leave another half – separated by a door – empty, then, when we open the door, gas will spread evenly through both halves of the box. From the purely mathematical viewpoint, it is also possible that, vice versa, if we start with a gas which is uniformly spread through the box, then at some future moment of time, all the molecules will concentrate in one half of this box, while the other half will remain empty: the probability of this event is small but still positive.

However, what the physicists claim is that such a spontaneous separation is simply not possible.

Need to go beyond probabilities is in good accordance with common sense. The impossibility of events with very low probability may sound counter-intuitive, but it is actually in good accordance with common sense. Suppose that you flip a coin – which you believe to be fair – several times, and every time it falls heads. If this happens two, three, even ten times in a row, you may still continue believing that the coin is fair and that the actual probability of heads is indeed 1/2. However, what if this happens 30 times in a row? 100 times in a row? Different people may have different thresholds, but for any person, there is some number after which the person will be absolutely sure that this coin is not fair.

Let us give another example. In each state lottery, usually, someone wins the big prize. If the same person wins the big prize two years in a row, one may still claim that this was a random coincidence. But what if the same person wins three years in a row? four years in a row? No matter how much you originally believe in the fairness of the state lottery process, if this continues year after year, eventually, every person will be convinced that the lottery is rigged.

Events with very small probability are not possible: can we describe this physical idea in purely probabilistic terms? We have mentioned that both physics and common sense use a rule that events with very small probability cannot happen. How can we describe this rule in precise terms?

At first glance, it may seem that we can describe this rule in purely probabilistic terms: namely, we can set up some threshold small value $p_0 \ll 1$, and we can claim that any event with probability $\leq p_0$ is not possible. However, a simple example of coin tossing shows that proposal does not work. Indeed, what we want to claim is that after tossing a coin a large number (N) of times, we cannot have a sequence HH...H of all heads. The probability of this event is 2^{-N}, which, for large N, is indeed a very small number. So, at first glance, it may seem that if we take $p_0 \geq 2^{-N}$, then we will be able to make the desired conclusion.

But the situation is not so easy. The problem is that *any* sequence of N heads and tails – including the actual sequence that we will get after tossing a coin N times – has the exact same probability 2^{-N}. So, if we require that no event with probability $\leq p_0$ is possible, we come up with a strange conclusion that no sequence of heads and tails is possible at all – which makes no sense, since, of course, we can flip the coin N times and record the results.

Comment. It is worth mentioning that there is a direct relation between this discussion and the notions of Algorithmic Information Theory, such as algorithmic randomness and Kolmogorov complexity; see, e.g., [22]. The main difference, however, is that the notion of algorithmic randomness is based on the assumption that events with probability 0 cannot occur, while we are trying to describe a more general statement: that not only events with zero probability cannot occur, but events with a sufficiently small positive probability cannot occur either.

Additional information is needed. The above simple example shows that we cannot separate possible from impossible events by only using the known probabilities

of different events. Thus, to properly describe physicists' reasoning (and our common sense), we need to supplement the probabilistic information with an additional information about what is possible.

How to describe what is possible. Let U be the set of all theoretically possible events. We assume that we know the probabilities of different events, i.e., that for some subsets $S \subseteq U$, we know the probability $p(S)$ that the actual event will be in S.

From all possible events, a physicist select a subset T of all events which are possible. The main idea that we want to formalize is that if the probability is very small, then the corresponding event is not possible. What constitutes "very small" depends on the situation, but it is clear that if we have a definable sequence of events $A_1 \supseteq A_2 \supseteq \ldots \supset A_n \supseteq \ldots$, with $p(A_n) \to 0$, then for some sufficiently large N, the probability of the corresponding event A_N becomes so small that this event is impossible, i.e., $T \cap A_N = \emptyset$.

This is what we trying to describe for the case of coin tossing: A_n is the event when the heads appear in the first n coin tosses; then, $p(A_n) = 2^{-n} \to 0$.

In general, we arrive at the following formalization:

Definition 4.1. [8] *Let U be a set with a probability measure p. We say that a subset $T \subseteq U$ is* a set of possible elements *if for every definable sequence A_n for which $A_n \supseteq A_{n+1}$ and $p(A_n) \to 0$, there exists N for which $T \cap A_N = \emptyset$.*

Need to go beyond probabilities. Sometimes, physicists use similar arguments even in situations when we do not know the probabilities. For example, physicists often expand a dependence in Taylor series $f(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \ldots$ When x is small, i.e., when $|x| \le \delta$ for some small δ, they argue that we can safely ignore quadratic (and higher order) terms in this expansion and assume that $f(x) \approx a_0 + a_1 \cdot x$; see, e.g., [5].

This conclusion is definitely justified if we know the value a_2, or, at least, if we know some a priori bound C on this value. Then, $|a_2 \cdot x^2| \le C \cdot \delta^2$, so when δ is sufficiently small, this term can indeed be safely ignored. However, physicists make this conclusion even when we do not know of any a priori bound on a_2. Their idea is that values which are too large are highly improbable.

In this case, we also have a series of events $A_n \supseteq A_{n+1}$: namely, A_n is the set of situations in which $|a_2| > n$. Here, we do not have probabilities, but we know that $\cap A_n = \emptyset$. Thus, no matter what is the (unknown) probability measure p, we have $p(A_n) \to 0$. As a result, we can use Definition 4.1 and conclude that for a sufficiently large N, events from A_N are impossible – hence $|a_2| \le N$.

Such situations lead to the following alternative definition that can be even when we do not know probabilities; see, e.g., [6, 7, 8, 15, 16, 17, 18, 19, 20]:

Definition 4.2. *Let U be a set. We say that a subset $T \subseteq U$ is* a set of possible elements *if for every definable sequence A_n for which $A_n \supseteq A_{n+1}$ and $\cap A_n = \emptyset$, there exists N for which $T \cap A_N = \emptyset$.*

What we do in this section. In this section, we show that many problems become algorithmically decidable if restrict ourselves to physically possible objects.

In general, many problems are not algorithmically decidable. In general, many computational problems are not algorithmically decidable; see, e.g., [2, 30]. As a simple example, let us consider the problem of deciding whether two given real numbers are equal or not.

In this problem, the input consists of two real numbers, and the desired output is "yes" or "no", depending on whether these numbers are equal or not.

To describe this problem in precise terms, we need to formulate how exactly we present the input to a computer. In practice, real numbers come from measurements, and measurements are never absolutely accurate. In principle, we can measure a real number x with higher and higher accuracy (if not now, then in the future). For example, for any integer n, we can measure this number with the accuracy of n binary digits, i.e., with the accuracy of 2^{-n}. As a result of each such measurement, we get a rational number r_n for which $|x - r_n| \le 2^{-n}$. This is exactly the usual definition of a computable real number: it is a process (maybe algorithmic, maybe involving measurements) that enables us, given an integer n, to generate a rational number r_n for which $|x - r_n| \le 2^{-n}$ [2, 30].

Computing with computable real numbers means that, in addition to usual computational steps, we can also generate some n, get the corresponding value r_n, and then use this value is computations.

Some things can be computed this way. For example, if we know computable real numbers x and y, then their sum $z = x + y$ is also a computable real number. Indeed, to compute the 2^{-n}-approximation t_n to the sum z, it is sufficient to take the sum $s_n = r_{n+1} + s_{n+1}$ of $2^{-(n+1)}$-approximations r_{n+1} and s_{n+1} to x and y. Indeed, from $|x - r_{n+1}| \le 2^{-(n+1)}$ and $|y - s_{n+1}| \le 2^{-(n+1)}$, we can then conclude that

$$|z - s_n| = |(x+y) - (x_{n+1} + y_{n+1})| = |(x - x_{n+1}) + (y - s_{n+1})| \le$$

$$|x - x_{n+1}| + |y - s_{n+1}| \le 2^{-(n+1)} + 2^{-(n+1)} = 2^{-n}.$$

However, it is not possible to algorithmically check whether the two computable numbers x and y are equal or not. Indeed, if this was possible, then, e.g., for equal real numbers $x = y = 0$ for which $r_n = s_n = 0$ for all n, our procedure will return the answer "yes". This procedure consists of finitely many steps, thus it can only ask for finitely many values r_n and s_n. Let N be the smallest number which is larger than all the requests n. Then, we can keep the same x, but take instead a different $y' = 2^{-N} \ne 0$ for which $s'_1 = \ldots = s'_{N-1} = 0$ (so our equality-checking procedure will not notice the difference), but $s'_N = s'_{N+1} = \ldots = 2^{-N}$. Since our procedure cannot notice the difference between y and y', it will still produce the same answer – that yes, the inputs are equal – while in reality, the new inputs $x = 0$ and $y' = 2^{-N} \ne 0$ are different.

Similar negative results are known for many other problems [2, 30].

If we restrict ourselves to possible pairs of real numbers, then equality becomes decidable. Let us show, following [17], that if we restrict ourselves to possible pairs (x, y), then it is algorithmically possible to check whether $x = y$ or $x \ne y$.

Indeed, the fact that we consider possible pairs of real numbers means that on the set $U = \mathbb{R} \times \mathbb{R}$ of all possible pairs of real numbers, we have a subset T of possible numbers that satisfied Definition 4.2. In particular, we can consider the following definable sequence of sets $A_n \stackrel{\text{def}}{=} \{(x,y) : 0 < |x-y| \le 2^{-n}\}$.

One can easily see that $A_n \supseteq A_{n+1}$ for all n and that $\cap A_n = \emptyset$. Thus, by Definition 4.2, there exists a natural number N for which $T \cap A_N = \emptyset$, i.e., for which no element $s \in T$ belongs to the set A_N. This, in turn, means that for every pair $(x,y) \in T$, either $|x-y| = 0$ (i.e., $x = y$) or $|x-y| > 2^{-N}$.

So, to check whether $x = y$ or not, it is sufficient to compute both x and y with accuracy $2^{-(N+2)}$, i.e., to compute values r_{N+2} and s_{N+2} for which $|x - r_{N+2}| \le 2^{-(N+2)}$ and $|y - s_{N+2}| \le 2^{-(N+2)}$. Then:

- if $x = y$, then, due to triangle inequality, we have

$$|r_{N+1} - s_{N+2}| \le |x - r_{N+2}| + |x - s_{N+2}| \le 2^{-(N+2)} + 2^{-(N+2)} = 2^{-(N+1)};$$

- on the other hand, if $x \ne y$, then from $|x-y| > 2^{-N}$, we conclude that

$$|r_{N+1} - s_{N+2}| \ge |x-y| - |x - r_{N+2}| - |y - s_{N+2}| >$$

$$2^{-N} - (2^{-(N+2)} + 2^{-(N+2)}) = 2^{-N} - 2^{-(N+1)} = 2^{-(N+1)}.$$

Thus, by checking whether $|r_{N+1} - s_{N+2}| \le 2^{-(N+1)}$ or whether $|r_{N+1} - s_{N+2}| > 2^{-(N+1)}$, we can decide whether $x = y$ or $x \ne y$.

Here, we compare rational numbers, i.e., ratios of integers, and for rational numbers, we can indeed algorithmically tell whether one is smaller or the other one is smaller.

Towards a general description of similar properties. To generalize the above result, let us come up with a general description of similar properties [12].

Let us start with reformulating the question of whether $x = y$ in generalizable terms. Specifically, we would like to describe the corresponding property in terms of the observable sequences r_n and s_n describing the real numbers x and y.

The equality between real numbers can indeed be described in these terms. Indeed, if $x = y$, then, for every n, we have

$$|r_n - s_n| \le |r_n - x| + |x - s_n| \le 2^{-n} + 2^{-n} = 2^{-(n-1)}.$$

Vice versa, let us assume that we have two computable real numbers x and y for which $|r_n - s_n| \le 2^{-(n-1)}$ for all n. In this case, due to $|x - r_n| \le 2^{-n}$ and $|y - s_n| \le 2^{-n}$, we have

$$|x-y| \le |x - r_n| + |r_n - s_n| + |s_n - y| \le 2^{-n} + 2^{-(n-1)} + 2^{-n} = 2^{-(n-2)}.$$

Since this hold for every n, for $n \to \infty$, we get $x = y$.

Thus, the equality between computable real numbers has the form

$$\forall n\,(|r_n - s_n| \le 2^{-(n-1)}).$$

In general, as shown, e.g., in [27, 30], many properties involving limits, differentiability, etc., can be described in similar terms, namely as an *arithmetic formula*

$$Qn_1\,Qn_2\ldots Qn_k\,F(r_1,\ldots,r_\ell,n_1,\ldots,n_k), \tag{4.1}$$

where:

- each Qn_i is either a universal quantifier $\forall n_i$ or an existential quantifier $\exists n_i$,
- $r_1,\ldots,r_\ell$ are corresponding sequences, and
- the property F is simply a propositional ("and", "or", and "not") combination of equalities and inequalities between the explicitly computable rational-valued expressions.

  In the above example of checking whether two given real numbers are equal:

- we have two sequences $\ell = 2$,
- we only have one quantifier $k = 1$,
- this quantifier is a universal quantifier $Q_1 = \forall$, and
- the property F has (in these terms) the form $|r_1(n_1) - r_2(n_1)| \le 2^{-(n_1-1)}$.

Let us show that for all such arithmetic expressions, the information on what is possible and what is not leads to algorithmic decidability.

Proposition 4.1. *For every arithmetic formula of type* (4.1) *and for every set T of possible tuples $r = (r_1,\ldots,r_\ell)$, there exists an algorithm that, given a tuple $r = (r_1,\ldots,r_\ell) \in T$, checks whether or not the given formula holds for this tuple.*

Conclusion. In this section, we have shown that in order to adequately describe physical reasoning, we need to supplement the usual probabilistic information with an additional knowledge describing what is physically possible and what is not. We have also shown that if we restrict ourselves to physically possible objects, then many problems become algorithmically decidable.

Conclusions

In this chapter, we showed that what is computable and what is feasibly computable depends:

- on what physical processes we allow, and
- on whether we are interested in:
  - general computations, in particular, solving mathematical problems, or
  - only processing physical data (in which case inputs must satisfy some physics-motivated constraints).

Somewhat surprisingly, the possibility to enhance computations comes not only when we consider specific physical models, but also when we take into account that, according to many physicists, no physical theory is perfect – i.e., no matter how well a theory fits the experimental data, it will eventually have to be modified.

Acknowledgements This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721.

Appendix: Proofs

A precise definition of definability.

Definition A1. *Let $\mathscr{L}$ be a theory, and let $P(x)$ be a formula from the language of the theory $\mathscr{L}$, with one free variable x for which the set $\{x\,|\,P(x)\}$ is defined in the theory $\mathscr{L}$. We will then call the set $\{x\,|\,P(x)\}$* $\mathscr{L}$-definable.

Crudely speaking, a set is $\mathscr{L}$-definable if we can explicitly *define* it in $\mathscr{L}$. The set of all real numbers, the set of all solutions of a well-defined equation, every set that we can describe in mathematical terms: all these sets are $\mathscr{L}$-definable.

This does not mean, however, that *every* set is $\mathscr{L}$-definable: indeed, every $\mathscr{L}$-definable set is uniquely determined by formula $P(x)$, i.e., by a text in the language of set theory. There are only denumerably many words and therefore, there are only denumerably many $\mathscr{L}$-definable sets. Since, e.g., in a standard model of set theory ZF, there are more than denumerably many sets of integers, some of them are thus not $\mathscr{L}$-definable.

Our objective is to be able to make mathematical statements about $\mathscr{L}$-definable sets. Therefore, in addition to the theory $\mathscr{L}$, we must have a stronger theory $\mathscr{M}$ in which the class of all $\mathscr{L}$-definable sets is a set – and it is a countable set.

Denotation. *For every formula F from the theory $\mathscr{L}$, we denote its Gödel number by $\lfloor F \rfloor$.*

Comment. A Gödel number of a formula is an integer that uniquely determines this formula. For example, we can define a Gödel number by describing what this formula will look like in a computer. Specifically, we write this formula in LaTeX, interpret every LaTeX symbol as its ASCII code (as computers do), add 1 at the beginning of the resulting sequence of 0s and 1s, and interpret the resulting binary sequence as an integer in binary code.

Definition A2. *We say that a theory $\mathscr{M}$ is* stronger *than $\mathscr{L}$ if it contains all formulas, all axioms, and all deduction rules from $\mathscr{L}$, and also contains a special predicate* $\mathrm{def}(n,x)$ *such that for every formula $P(x)$ from $\mathscr{L}$ with one free variable, the formula $\forall y\,(\mathrm{def}(\lfloor P(x)\rfloor, y) \leftrightarrow P(y))$ is provable in $\mathscr{M}$.*

The existence of a stronger theory can be easily proven: indeed, for $\mathscr{L}$=ZF, there exists a stronger theory $\mathscr{M}$. As an example of such a stronger theory, we can simply take the theory $\mathscr{L}$ plus all countably many equivalence formulas as described in Definition A2 (formulas corresponding to all possible formulas $P(x)$ with one free variable). This theory clearly contains $\mathscr{L}$ and all the desired equivalence formulas, so all we need to prove is that the resulting theory $\mathscr{M}$ is consistent (provided that $\mathscr{L}$ is consistent, of course). Due to compactness principle, it is sufficient to prove that

for an arbitrary finite set of formulas $P_1(x),\dots,P_m(x)$, the theory $\mathscr{L}$ is consistent with the above reflection-principle-type formulas corresponding to these properties $P_1(x),\dots,P_m(x)$.

This auxiliary consistency follows from the fact that for such a finite set, we can take

$$\mathrm{def}(n,y) \leftrightarrow (n = \lfloor P_1(x)\rfloor \,\&\, P_1(y)) \vee \dots \vee (n = \lfloor P_m(x)\rfloor \,\&\, P_m(y)).$$

This formula is definable in $\mathscr{L}$ and satisfies all m equivalence properties. The statement is proven.

Important comments. In the main text, we will assume that a theory $\mathscr{M}$ that is stronger than $\mathscr{L}$ has been fixed; proofs will mean proofs in this selected theory $\mathscr{M}$.

An important feature of a stronger theory $\mathscr{M}$ is that the notion of an $\mathscr{L}$-definable set can be expressed within the theory $\mathscr{M}$: a set S is $\mathscr{L}$-definable if and only if

$$\exists n \in \mathbb{N}\, \forall y (\mathrm{def}(n,y) \leftrightarrow y \in S).$$

In the paper, when we talk about definability, we will mean this property expressed in the theory $\mathscr{M}$. So, all the statements involving definability become statements from the theory $\mathscr{M}$ itself, *not* statements from metalanguage.

Proof of Proposition 3.1. Let us fix an integer C. To prove the desired property for this C, let us prove that the set T of all the sequences which do not satisfy this property, i.e., for which $K(\alpha_1\dots\alpha_n \,|\, \omega) \ge K(\alpha_1\dots\alpha_n) - C$ for all n, is a physical theory in the sense of Definition 1. For this, we need to prove that this set T is non-empty, closed, nowhere dense, and definable. Then, from Definition 2, it will follow that the sequence ω does not belong to this set and thus, that the conclusion of Proposition 1 is true.

The set T is clearly non-empty: it contains, e.g., a sequence $\omega = 00\dots0\dots$ which does not affect computations. The set T is also clearly definable: we have just defined it.

Let us prove that the set T is closed. For that, let us assume that $\omega^{(m)} \to \omega$ and $\omega^{(m)} \in T$ for all m. We then need to prove that $\omega \in T$. Indeed, let us fix n, and let us prove that $K(\alpha_1\dots\alpha_n \,|\, \omega) \ge K(\alpha_1\dots\alpha_n) - C$. We will prove this by contradiction. Let us assume that $K(\alpha_1\dots\alpha_n \,|\, \omega) < K(\alpha_1\dots\alpha_n) - C$. This means that there exists a program p of length $\mathrm{len}(p) < K(\alpha_1\dots\alpha_n) - C$ which uses ω to compute $\alpha_1\dots\alpha_n$. This program uses only finitely many bits of ω; let B be the largest index of these bits. Due to $\omega^{(m)} \to \omega$, there exists M for which, for all $m \ge M$, the first B bits of $\omega^{(m)}$ coincide with the first B bits of the sequence ω. Thus, the same program p will work exactly the same way – and generate the sequence $\alpha_1\dots\alpha_n$ – if we use $\omega^{(m)}$ instead of ω. But since $\mathrm{len}(p) < K(\alpha_1\dots\alpha_n) - C$, this would means that the shortest length $K(\alpha_1\dots\alpha_n \,|\, \omega^{(m)})$ of all the programs which use $\omega^{(m)}$ to compute $\alpha_1\dots\alpha_n$ also satisfies the inequality $K(\alpha_1\dots\alpha_n \,|\, \omega^{(m)}) < K(\alpha_1\dots\alpha_n) - C$. This inequality contradicts to our assumption that $\omega^{(m)} \in T$ and thus, that $K(\alpha_1\dots\alpha_n \,|\, \omega^{(m)}) \ge K(\alpha_1\dots\alpha_n) - C$. The contradiction proves that the set T is indeed closed.

Let us now prove that the set T is nowhere dense, i.e., that for every finite sequence $\omega_1\dots\omega_m$, there exists a continuation ω which does not belong to the set T.

Indeed, as such a continuation, we can simply take a sequence $\omega = \omega_1 \ldots \omega_m \alpha_1 \alpha_2 \ldots$ obtained by appending α at the end. For this new sequence, computing $\alpha_1 \ldots \alpha_n$ is straightforward: we just copy the values α_i from the corresponding places of the new sequence ω. Here, the relative Kolmogorov complexity $K(\alpha_1 \ldots \alpha_n \,|\, \omega)$ is very small and is, thus, much smaller than the complexity $K(\alpha_1 \ldots \alpha_n)$ which – since ZF is not decidable – grows fast with n.

The proposition is proven.

Proof of Proposition 3.2.

1°. As the desired ph-algorithm, we will, given an instance i, simply produce the result ω_i of the i-th experiment. Let us prove, by contradiction, that this algorithm satisfies the desired property.

2°. We want to prove that for every $\varepsilon > 0$ and for every n, there exists an integer $N \geq n$ for which

$$\#\{i \leq N : i \in S_{\mathscr{P}} \,\&\, \omega_i = s_{\mathscr{P},i}\} > (1-\varepsilon) \cdot \#\{i \leq N : i \in S_{\mathscr{P}}\}.$$

The assumption that this property is not satisfied means that for some $\varepsilon > 0$ and for some integer n, we have

$$\#\{i \leq N : i \in S_{\mathscr{P}} \,\&\, \omega_i = s_{\mathscr{P},i}\} \leq (1-\varepsilon) \cdot \#\{i \leq N : i \in S_{\mathscr{P}}\} \text{ for all } N \geq n. \quad (3.1)$$

Let T denote the set of all the sequences x that satisfy the property (3.1), i.e., let

$$T \stackrel{\text{def}}{=} \{x : \#\{i \leq N : i \in S_{\mathscr{P}} \,\&\, x_i = s_{\mathscr{P},i}\} \leq (1-\varepsilon) \cdot \#\{i \leq N : i \in S_{\mathscr{P}}\} \text{ for all } N \geq n\}.$$

We will prove that this set T is a physical theory in the sense of Definition 3.1.

Then, due to Definition 3.2 and the fact that the sequence ω satisfies the no-perfect-theory principle, we will be able to conclude that $\omega \notin T$, and thus, that the property (3.1) is not satisfied for the given sequence ω. This will conclude the proof by contradiction.

3°. By definition of a physical theory T, it is a set which is non-empty, closed, nowhere dense, and definable. Let us prove these four properties one by one.

3.1°. Non-emptiness comes from the fact that the sequence x_i for which $x_i = \neg s_{\mathscr{P},i}$ for $i \in S_{\mathscr{P}}$ and $x_i = 0$ otherwise clearly belongs to this set: for this sequence, for every N, we have

$$\#\{i \leq N : i \in S_{\mathscr{P}} \,\&\, x_i = s_{\mathscr{P},i}\} = 0$$

and thus, the desired property is satisfied.

3.2°. Let us prove that the set T is closed, i.e., that if we have a family of sequences $x^{(m)} \in T$ for which $x^{(m)} \to \omega$, then $x \in T$.

Indeed, let us take any $N \neq n$, and let us prove that

$$\#\{i \leq N : i \in S_{\mathscr{P}} \,\&\, x_i = s_{\mathscr{P},i}\} \leq (1-\varepsilon) \cdot \#\{i \leq N : i \in S_{\mathscr{P}}\}$$

for this N. Due to $x^{(m)} \to x$, there exists M for which, for all $m \geq M$, the first N bits of $x^{(m)}$ coincide with the first N bits of the sequence x: $x_i^{(m)} = \omega_i$ for all $i \leq N$. Thus,

$$\#\{i \leq N : i \in S_{\mathscr{P}} \,\&\, x_i = s_{\mathscr{P},i}\} = \#\{i \leq N : i \in S_{\mathscr{P}} \,\&\, x_i^{(m)} = s_{\mathscr{P},i}\}.$$

Since $x^{(m)} \in T$, we have

$$\#\{i \leq N : i \in S_{\mathscr{P}} \,\&\, x_i^{(m)} = s_{\mathscr{P},i}\} \leq (1-\varepsilon) \cdot \#\{i \leq N : i \in S_{\mathscr{P}}\},$$

thus

$$\#\{i \leq N : i \in S_{\mathscr{P}} \,\&\, x_i = s_{\mathscr{P},i}\} \leq (1-\varepsilon) \cdot \#\{i \leq N : i \in S_{\mathscr{P}}\}.$$

So, the set T is indeed closed.

3.3°. Let us now prove that the set T is nowhere dense, i.e., that for every finite sequence $x_1 \ldots x_m$, there exists a continuation x which does not belong to the set T.

Indeed, as such a continuation, we can simply take a sequence

$$x = x_1 \ldots x_m x_{m+1} x_{m+2} \ldots$$

where for $i > m$, we take $x_i = s_{\mathscr{P},i}$ if $i \in S_{\mathscr{P}}$ and $x_i = 0$ otherwise. For this new sequence, for every N, at most m first instances may lead to results different from $s_{\mathscr{P},i}$, so we have

$$\#\{i \leq N : i \in S_{\mathscr{P}} \,\&\, x_i = s_{\mathscr{P},i}\} \geq \#\{i \leq N : i \in S_{\mathscr{P}}\} - m.$$

When $N \to \infty$, then $\#\{i \leq N : i \in S_{\mathscr{P}}\} \to \infty$, so for sufficiently large N, we have

$$\#\{i \leq N : i \in S_{\mathscr{P}}\} - m > (1-\varepsilon) \cdot \#\{i \leq N : i \in S_{\mathscr{P}}\},$$

thus,

$$\#\{i \leq N : i \in S_{\mathscr{P}} \,\&\, x_i = s_{\mathscr{P},i}\} > (1-\varepsilon) \cdot \#\{i \leq N : i \in S_{\mathscr{P}}\},$$

and we cannot have

$$\#\{i \leq N : i \in S_{\mathscr{P}} \,\&\, x_i = s_{\mathscr{P},i}\} \leq (1-\varepsilon) \cdot \#\{i \leq N : i \in S_{\mathscr{P}}\}.$$

Therefore, this continuation does not belong to the set T.

3.4°. Finally, since the formula (3.1) explicitly defines the set T, this set T is clearly definable.

So, T is a physical theory, hence $\omega \notin T$, and the proposition is proven.

Proof of Proposition 3.3. Let us assume that P≠NP. We then need to prove that for every feasible ph-algorithm $\mathscr{A}$, it is not possible to have

$$\#\{i \leq N : i \in S_{\mathscr{P}} \,\&\, \mathscr{A}(\omega, i) = s_{\mathscr{P},i}\} = \#\{i \leq N : i \in S_{\mathscr{P}}\}$$

for all natural numbers N.

To prove this impossibility, let us consider, for each feasible ph-algorithm $\mathscr{A}$, the set

$$T(\mathscr{A}) \stackrel{\text{def}}{=} \{x : \#\{i \le N : i \in S_{\mathscr{P}} \,\&\, \mathscr{A}(x,i) = s_{\mathscr{P},i}\} = \#\{i \le N : i \in S_{\mathscr{P}}\} \text{ for all } N\}.$$

Similarly to the proof of Proposition 3.2, we can show that this set $T(\mathscr{A})$ is closed and definable.

Let us prove that the set $T(\mathscr{A})$ is nowhere dense, i.e., that for every finite sequence $x_1 \ldots x_m$, there exists a continuation x which does not belong to the set $T(\mathscr{A})$. Indeed, we can simply extend the original finite sequence $x_1 \ldots x_m$ by 0s. In this case, when the oracle has only finitely many nonzero bits, we can incorporate these bits into an algorithm and get a feasible non-oracle algorithm $\mathscr{A}'$ which produces the same results: $\mathscr{A}'(i) = \mathscr{A}(x,i)$ for all i.

Let us prove, by contradiction, that $x \notin T(\mathscr{A})$. Indeed, if $x \in T(\mathscr{A})$, this would mean that

$$\#\{i \le N : i \in S_{\mathscr{P}} \,\&\, \mathscr{A}'(i) = s_{\mathscr{P},i}\} = \#\{i \le N : i \in S_{\mathscr{P}}\}$$

for all N. Thus, the feasible non-oracle algorithm $\mathscr{A}'$ solves all the instances of the original NP-complete problem $\mathscr{P}$, which contradicts to our assumption that P$\neq$NP. This contradiction proves that $x \notin T(\mathscr{A})$ and thus, the set $T(\mathscr{A})$ is indeed nowhere dense.

We have thus proven that the set $T(\mathscr{A})$ is closed, nowhere dense, and definable. The only property which is still missing from the definition of a physical theory (Definition 3.1) is non-emptiness. We do not know whether the set $T(\mathscr{A})$ is non-empty or not, but we can prove the desired impossibility in both cases.

If the set $T(\mathscr{A})$ is non-empty, then this set is a theory in the sense of Definition 1, and thus, since the sequence ω satisfies the no-perfect-theory principle, we have $\omega \notin T(\mathscr{A})$. This means that the ph-algorithm $\mathscr{A}$ is not solving all instances of the problem $\mathscr{P}$.

If the set $T(\mathscr{A})$ is empty, this also means that the ph-algorithm $\mathscr{A}$ does not solve all instances of the problem $\mathscr{P}$ – no matter what oracle we use.

The proposition is proven.

Proof of Proposition 3.4. Let us assume that no non-oracle feasible algorithm δ-solves the problem $\mathscr{P}$. We then need to prove that for every feasible ph-algorithm $\mathscr{A}$, it is not possible to have N_0 for which

$$\#\{i \le N : i \in S_{\mathscr{P}} \,\&\, \mathscr{A}(\omega,i) = s_{\mathscr{P},i}\} > \delta \cdot \#\{i \le N : i \in S_{\mathscr{P}}\}$$

for all natural numbers $N \ge N_0$.

To prove this impossibility, let us consider, for each feasible ph-algorithm $\mathscr{A}$ and for each natural number N_0, the set

$$T(\mathscr{A}, N_0) \stackrel{\text{def}}{=}$$

$$\{x : \#\{i \le N : i \in S_{\mathscr{P}} \,\&\, \mathscr{A}(x,i) = s_{\mathscr{P},i}\} > \delta \cdot \#\{i \le N : i \in S_{\mathscr{P}}\} \text{ for all } N \ge N_0\}.$$

Similarly to the proof of Proposition 3.2, we can show that this set $T(\mathscr{A},N_0)$ is closed and definable.

Let us prove that the set $T(\mathscr{A},N_0)$ is nowhere dense, i.e., that for every finite sequence $x_1\ldots x_m$, there exists a continuation x which does not belong to the set $T(\mathscr{A},N_0)$. Indeed, similarly to the proof of Proposition 3.3, we can extend the original finite sequence $x_1\ldots x_m$ by 0s. In this case, when the oracle has only finitely many nonzero bits, we can incorporate these bits into an algorithm and get a feasible non-oracle algorithm $\mathscr{A}'$ which produces the same results: $\mathscr{A}'(i)=\mathscr{A}(x,i)$ for all i.

Let us prove, by contradiction, that $x\notin T(\mathscr{A},N_0)$. Indeed, if $x\in T(\mathscr{A},N_0)$, this would mean that

$$\#\{i\le N: i\in S_{\mathscr{P}}\,\&\,\mathscr{A}'(i)=s_{\mathscr{P},i}\}>\delta\cdot\#\{i\le N: i\in S_{\mathscr{P}}\}$$

for all $N\ge N_0$. Thus, the feasible non-oracle algorithm $\mathscr{A}'$ δ-solves the original NP-complete problem $\mathscr{P}$, which contradicts to our assumption that no such feasible non-oracle algorithm is possible. This contradiction proves that $x\notin T(\mathscr{A},N_0)$ and thus, the set $T(\mathscr{A},N_0)$ is indeed nowhere dense.

We have thus proven that the set $T(\mathscr{A},N_0)$ is closed, nowhere dense, and definable. The only property which is still missing from the definition of a physical theory (Definition 3.1) is non-emptiness. We do not know whether the set $T(\mathscr{A},N_0)$ is non-empty or not, but we can prove the desired impossibility in both cases.

For each N_0, if the set $T(\mathscr{A},N_0)$ is non-empty, then this set is a theory in the sense of Definition 3.1, and thus, since the sequence ω satisfies the no-perfect-theory principle, we have $\omega\notin T(\mathscr{A},N_0)$, i.e.,

$$\#\{i\le N: i\in S_{\mathscr{P}}\,\&\,\mathscr{A}(\omega,i)=s_{\mathscr{P},i}\}\le\delta\cdot\#\{i\le N: i\in S_{\mathscr{P}}\}\text{ for some }N\ge N_0.\quad(3.2)$$

If the set $T(\mathscr{A},N_0)$ corresponding to a given N_0 is empty, then also $\omega\notin T(\mathscr{A},N_0)$, i.e., we also have the property (3.2).

Since the property (3.2) holds for all N_0, this means that the ph-algorithm $\mathscr{A}$ does not δ-solve the problem $\mathscr{P}$.

The proposition is proven.

Proof of Proposition 4.1. If the formula (4.1) had no quantifiers, then we could simply plug in the corresponding values into this formula and check whether the corresponding formula holds or not. The problem is with the quantifiers: while we can easily check whether some property holds for a specific value n_i, it is not possible to directly check whether this property holds for all infinitely many natural numbers $n_i=0,1,2,\ldots$ The situation would be different if we could have a bound N on possible values of n_i, i.e., if the quantifier had the form $\forall n_i\le N$ or $\exists n_i\le N$: in this case, we can simply test all possible values $n_i\le N$.

Let us show that for tuples from the set T, we can indeed have such bounds on the variables n_i. Let us start with a bound on n_1. For the variable n_1, there are two possible cases: when Q_1 is a universal quantifier and when Q_1 is an existential quantifier. Let us consider these two cases one by one.

In the first case, the formula (4.1) has the form $\forall n_1\, G(n_1)$, for some expression $G(n_1)$ (with one fewer quantifier). Let us take

$$A_n = \{r : \forall n_1\, (n_1 \le n \to G(n_1)) \,\&\, \neg\forall n_1\, G(n_1)\}.$$

One can easily check that $A_n \supseteq A_{n+1}$ and $\cap A_n = \emptyset$. Thus, there exists a natural number N for which $T \cap A_N = \emptyset$. So, for $r \in T$, if $\forall n_1\, (n_1 \le N \to G(n_1))$, we cannot have $\neg\forall n_1\, G(n_1)$, so we must have $\forall n_1\, G(n_1)$. Clearly, $\forall n_1\, G(n_1)$ always implies $\forall n_1\, (n_1 \le N \to G(n_1))$. Thus, for $r \in T$, $\forall n_1\, G(n_1)$ with an unlimited quantifier is equivalent to a formula $\forall n_1\, (n_1 \le N \to G(n_1))$ with a bounded quantifier.

In the second case, the formula (4.1) has the form $\exists n_1\, G(n_1)$, for some expression G (with one fewer quantifier). Let us take

$$A_n = \{r : \neg\exists n_1\, (n_1 \le n \,\&\, G(n_1)) \,\&\, \exists n_1\, G(n_1)\}.$$

One can easily check that $A_n \supseteq A_{n+1}$ and $\cap A_n = \emptyset$. Thus, there exists a natural number N for which $T \cap A_N = \emptyset$. So, for $r \in T$, if $\neg\exists n_1\, (n_1 \le N \,\&\, G(n_1))$, we cannot have $\exists n_1\, G(n_1)$, so we must have $\neg\exists n_1\, G(n_1)$. Clearly, $\neg\exists n_1\, G(n_1)$ always implies $\neg\exists n_1\, (n_1 \le N \,\&\, G(n_1))$. Thus, for $r \in T$, $\neg\exists n_1\, G(n_1)$ is equivalent to $\neg\exists n_1\, (n_1 \le N \,\&\, G(n_1))$. So, by taking negations, we conclude that the original formula $\exists n_1\, G(n_1)$ with an unlimited quantifier is equivalent to a formula $\exists n_1\, (n_1 \le N \,\&\, G(n_1))$ with a bounded quantifier.

Now, we have reduced the original formula with k quantifiers to a formula in which the first quantifier is bounded. This bounded-quantifier formula is equivalent to, correspondingly, $G(0) \,\&\, G(1) \,\&\, \ldots \,\&\, G(N)$ or to $G(0) \vee G(1) \vee \ldots \vee G(N)$, where the corresponding formulas $G(n_1)$ have $k-1$ quantifiers. So, if we can find the truth values of each of these (finitely many) formulas $G(n_1)$, we could be able to check the truth value of the original formula (4.1).

For each of these formulas $G(n_1)$ with $k-1$ quantifiers, we can apply the same reduction to reduce them to formulas with $k-2$ quantifiers, etc., until we get formulas with no quantifiers at all – which can be therefore directly checked.

This reduction proves that it is indeed algorithmically possible to check whether a given formula (4.1) holds or not for a given tuple r. The proposition is proven.

References

1. Aaronson, S.: NP-complete problems and physical reality, SIGACT News **36**, 30–52 (2005)
2. Bishop, E.: Foundations of Constructive Analysis, McGraw-Hill, New York (1967)
3. Boltzmann, L.: Bemrkungen über einige Probleme der mechanischen Wärmtheorie, Wiener Ber. II, **75**, 62–100 (1877)
4. Feynman, R.P.: Statistical Mechanics, W. A. Benjamin, New York (1972)
5. Feynman, R., Leighton, R., Sands, M.: The Feynman Lectures on Physics, Addison Wesley, Boston, Massachusetts (2005)
6. Finkelstein, A.M., Kosheleva, O., Kreinovich, V., Starks, S.A., Nguyen, H.T.: To properly reflect physicists' reasoning about randomness, we also need a maxitive (possibility) mea-

sure, In: Proceedings of the 2005 IEEE International Conference on Fuzzy Systems FUZZ-IEEE'2005, Reno, Nevada, May 22–25, 2005, pp. 1044–1049 (2015)
7. Finkelstein, A.M., Kosheleva, O., Kreinovich, V., Starks, S.A., Nguyen, H.T.: Use of maxitive (possibility) measures in foundations of physics and description of randomness: case study, In: Proceedings of the 24th International Conference of the North American Fuzzy Information Processing Society NAFIPS'2005, Ann Arbor, Michigan, June 22–25, 2005, pp. 687–692 (2005)
8. Finkelstein, A.M., Kreinovich, V.: Impossibility of hardly possible events: physical consequences, In: Abstracts of the 8th International Congress on Logic, Methodology and Philosophy of Science, Moscow **5**(2), 25–27 (1987)
9. Jalal-Kamali, A., Nebesky, O., Durcholz, M.H., Kreinovich, V., Longpré, L.: Towards a "generic" notion of genericity: from 'typical' and 'random' to meager, shy, etc., Journal of Uncertain Systems **6**(2), 104–113 (2012)
10. Koshelev, M., Kreinovich, V.: Towards computers of generation omega – non-equilibrium thermodynamics, granularity, and acausal processes: a brief survey, In: Proceedings of the International Conference on Intelligent Systems and Semiotics (ISAS'97), pp. 383–388, National Institute of Standards and Technology Publ., Gaithersburg, Maryland (1997)
11. Kosheleva, O.M., Kreinovich, V.: What can physics give to constructive mathematics, In: Mathematical Logic and Mathematical Linguistics, pp. 117–128, Kalinin (1981, in Russian)
12. Kosheleva, O., Kreinovich, V.: Adding possibilistic knowledge to probabilities makes many problems algorithmically decidable, Proceedings of the World Congress of the International Fuzzy Systems Association IFSA'2015, joint with the Annual Conference of the European Society for Fuzzy Logic and Technology EUSFLAT'2015, Gijon, Asturias, Spain, June 30 – July 3, 2015, to appear (2015)
13. Kosheleva, O.M., Soloviev, S.V.: On the logic of using observable events in decision making, Proceedings of the IX National USSR Symposium on Cybernetics, Moscow, 1981, pp. 49–51 (1981, in Russian).
14. Kosheleva, O., Zakharevich, M., Kreinovich, V.: If many physicists are right and no physical theory is perfect, then by using physical observations, we can feasibly solve almost all instances of each NP-complete problem, Mathematical Structures and Modeling **31**, 4–17 (2014)
15. Kreinovich, V.: Toward formalizing non-monotonic reasoning in physics: the use of Kolmogorov complexity and Algorithmic Information Theory to formalize the notions "typically" and "normally", In: Sheremetov, L., Alvarado, M.(eds.), Proceedings of the Workshops on Intelligent Computing WIC'04 associated with the Mexican International Conference on Artificial Intelligence MICAI'04, Mexico City, Mexico, April 26–27, 2004, pp. 187–194 (2004)
16. Kreinovich, V.: Toward formalizing non-monotonic reasoning in physics: the use of Kolmogorov complexity, Revista Iberoamericana de Inteligencia Artificial **41**, 4–20 (2009)
17. Kreinovich, V.: Negative results of computable analysis disappear if we restrict ourselves to random (or, more generally, typical) inputs, Mathematical Structures and Modeling **25**, 100–113 (2012)
18. Kreinovich, V.: Towards formalizing non-monotonic reasoning in physics: logical approach based on physical induction and its relation to Kolmogorov complexity. In Erdem, E., Lee, J., Lierler, Y., Pearce, D. (eds.) Correct Reasoning: Essays on Logic-Based AI in Honor of Vladimir Lifschitz, Springer Verlag, Lectures Notes in Computer Science **7265**, 390–404 (2012)
19. Kreinovich, V., Finkelstein, A.M.: Towards applying computational complexity to foundations of physics, Notes of Mathematical Seminars of St. Petersburg Department of Steklov Institute of Mathematics **316**, 63–110 (2004); reprinted in Journal of Mathematical Sciences **134**(5), 2358–2382 (2006)
20. Kreinovich, V., Kosheleva, O.: Logic of scientific discovery: how physical induction affects what is computable, In: Proceedings of the International Interdisciplinary Conference Philosophy, Mathematics, Linguistics: Aspects of Interaction 2014 PhML'2014, St. Petersburg, Russia, April 21–25, 2014, pp. 116–127 (2014)

21. Kreinovich, V., Margenstern, M.: In some curved spaces, one can solve NP-hard problems in polynomial time, Notes of Mathematical Seminars of St. Petersburg Department of Steklov Institute of Mathematics **358**, 224–250 (2008); reprinted in Journal of Mathematical Sciences **158**(5), 727–740 (2009)
22. Li, M., Vitányi, P.M.B.: An Introduction to Kolmogorov Complexity, Springer-Verlag, Berlin, Heidelberg, New York (2008)
23. Misner, C.W., Thorne, K.S. Wheeler, J.A.: Gravitation, W. H. Freeman and Company, San Francisco (1973)
24. Morgenstein, D., Kreinovich, V.: Which algorithms are feasible and which are not depends on the geometry of space-time, Geombinatorics **4**(3), 80–97 (1995)
25. Oxtoby, J.C.: Measure and Category: A Survey of the Analogies between Topological and Measure Spaces, Springer Verlag, New York, Heidelberg, Berlin (1980)
26. Papadimitriou, C.: Computational Complexity, Addison Welsey, Reading, Massachusetts (1994)
27. Rogers, H. Jr.: The Theory of Recursive Functions and Effective Computability, MIT Press, Cambridge, Massachusetts (1987)
28. Srikanth, R.: The quantum measurement problem and physical reality: a computation theoretic perspective, in: Goswami, D. (ed.) Quantum Computing: Back Action 2006, IIT Kanpur, India, March 2006, AIP Conference Proceedings **864**, 178–193 (2006)
29. Thorne, K.S.: From Black Holes to Time Warps: Einstein's Outrageous Legacy, W.W. Norton & Company, New York (1994)
30. Weihrauch, K.: Computable Analysis, Springer Verlag, Berlin (2000)
31. Zakharevich, M., Kosheleva, O.: If many physicists are right and no physical theory is perfect, then the use of physical observations can enhance computations, Journal of Uncertain Systems **8**(3), 227–232 (2014)

Index

"almost" black hole, 5

acausal processes, 6
algorithm
  exhaustive search, 3
Algorithmic Information Theory, 15
algorithmic randomness, 10, 15
arithmetic formula, 19

Baire first category, 10
black hole, 5

closed set, 9
computable, 1
  feasibly, 1
computable real number, 17
computation
  with an oracle, 12
computations
  parallel, 3
curved space-time, 5

data processing, 2
definable set, 8, 20

entropy, 14
Euclidean geometry, 4
Euclidean space, 4
exhaustive search, 3

feasible, 1

Gödel number, 20
geometry
  Euclidean, 4
  hyperbolic, 5
  Lobachevsky, 5
  non-Euclidean, 5
graph coloring, 3

hyperbolic geometry, 5
hyperbolic space, 5

irreversible process, 14

Kolmogorov complexity, 10, 15
Kolmorogov complexity
  relative, 11

Lobachevsky geometry, 5
Lobachevsky space, 5

meager set, 10

no-perfect-theory principle, 6
non-Euclidean geometry, 5
nowhere dense set, 9
NP, 3
NP-complete, 3, 11

oracle, 12

$P\stackrel{?}{=}NP$, 3
parallelization, 3
physical theory, 8
physics, 2
propositional satisfiability, 3

quantum physics, 6, 13

random sequence, 10
randomness, 10

set theory, 10

space-time
  curved, 5
speed of light, 4
statistical physics, 14

topology, 9

undecidable problems, 17

wormhole, 5

Zermelo-Fraenkel set theory, 10
ZF, 10, 20