Complexity vs energy: theory of computation and theoretical physics

Y I Manin

Max-Planck-Institut für Mathematik, Vivatsgasse 7, 53111 Bonn, Germany

Abstract. This paper is a survey based upon the talk at the satellite *QQQ* conference to ECM6, *3Quantum: Algebra Geometry Information*, Tallinn, July 2012. It is dedicated to the analogy between the notions of *complexity* in theoretical computer science and *energy* in physics. This analogy is not metaphorical: I describe three precise mathematical contexts, suggested recently, in which mathematics related to (un)computability is inspired by and to a degree reproduces formalisms of statistical physics and quantum field theory.

Contents

1. Introduction and summary

This survey of several recent papers [15, 17, 18, 20, 21] and [19], the latter can also be read in my preprint [arXiv:1301.0081], is dedicated to a deep analogy between the notions of *complexity* in theoretical computer science and *energy* in physics. The analogy is not metaphorical: we describe several precise mathematical contexts, suggested recently, in which mathematics related to (un)computability is inspired by (and to a degree reproduces) formalisms of statistical physics and quantum field theory.

Namely, after recalling basics of the classical computability theory in Sec. 2, we turn to three main subjects:

a) *The problem of (un)computability of the asymptotic bound for error-correcting codes over a fixed finite alphabet (Sec. 3).*

Here M. Marcolli and the author have shown in [21] based upon [15] that the asymptotic bound arises as a phase transition curve between different thermodynamic phases. The relevant partition function is a sum over the ensemble of all codes in which the role of energy is played by the Kolmogorov complexity of the code.

b) *The problem of mathematical foundation for the empirical Zipf's law (describing e.g. the frequency distributions of words in natural languages).*

It was suggested that this distribution reflects minimization of certain "effort". I show that (in certain contexts) if this effort is defined as complexity, Zipf's law emerges from L. Levin's *a priori* distributions, mathematical theory of which was founded in the 1970's: see Sec. 4 and more detailed argumentation in [20].

c) *The problem of uncomputable in the computability theory (Sec. 5-7).*

It is well-known that the theory of computability unavoidably leads to effects of uncomputability in its own realm: basically, it may be impossible to decide in finite time whether a partial recursive function is defined at a given point.

I draw an analogy between this and problems of infinities in perturbative Quantum Field Theory (QFT). Moreover, I suggest that renormalization schemes from QFT involving first a deformation of the problem and then "subtraction of infinities" can be fruitfully applied in computation theory. This procedure as well involves Kolmogorov complexity. The basic common elements of the two formalisms are graphs appearing as Feynman diagrams in QFT and as flowcharts in computation theory. For more details and related results cf [17, 18, 16].

There are no proofs in this report: we focus on the presentation of basic ideas. The paper is a survey based upon the talk at the satellite *QQQ* conference to ECM6, *3Quantum: Algebra Geometry Information*, Tallinn, July 2012.

2. A brief guide to computability: operadic and categorical perspective
Any single approach to mathematical notion of computability - Turing's machines, Church's lambda calculus, Markov's algorithms - by necessity bypasses rich intuitions governing other approaches. But since this is unavoidable, and since our goal here is to pave the shortest way to Kolmogorov's complexity, for us computability theory here will be based on the theory of (partial) recursive functions.

2.1. Three descriptions of partial recursive functions
A "function", say, $f : X \to Y$, below always means a pair $(f, D(f))$, where $D(f) \subset X$ and $f : D(f) \to Y$ a set-theoretic map. The definition domain is not always mentioned explicitly. If $D(f) = X$, the function might be called "total"; generally it may be called "partial" one. The other extremal case is that of "empty function", with $D(f) = \emptyset$. We put $\mathbf{Z}_+ := \{1, 2, 3, \dots\}$.

(i) *Intuitive description.* A function $f : \mathbf{Z}_+^m \to \mathbf{Z}_+^n$ is (partial) recursive iff it is "semi-computable" in the following sense: there exists an algorithm F accepting as inputs vectors $x = (x_1, \dots, x_m) \in \mathbf{Z}_+$ with the following properties:

    – if $x \in D(f)$, F produces as output $f(x)$.
    – if $x \notin D(f)$, F either produces the output "NO", or works indefinitely long without producing any output.

(ii) *Formal description (sketch).* It starts with two lists:

    – An explicit list of "obviously" semi-computable *basic functions* such as constant functions, projections onto i-th coordinate etc.
    – An explicit list of *elementary operations*, performed over functions, such as composition, inductive definition, and implicit definition by equation, that can be applied to several semi-computable functions and "obviously" produces from them a new semi-computable function.

After that, the set of *partial recursive functions* is defined as the minimal set of functions $f : \mathbf{Z}_+^m \to \mathbf{Z}_+^n$, with all $m, n \geq 0$, containing all basic functions and closed w.r.t. all elementary operations. For details, see e.g. [14], Ch. V.

(iii) *Diophantine description (a difficult theorem).* A function $f : \mathbf{Z}_+^m \to \mathbf{Z}_+^n$ is partial recursive iff there is a polynomial

$$P(x_1, \ldots, x_m; y_1, \ldots, y_n; t_1, \ldots, t_q) \quad \in \mathbf{Z}[x, y, t]$$

such that the graph

$$\Gamma_f := \{(x, f(x))\} \quad \subset \mathbf{Z}_+^m \times \mathbf{Z}_+^n$$

is the projection of the subset $P = 0$ in $\mathbf{Z}_+^m \times \mathbf{Z}_+^n \times \mathbf{Z}_+^q$. For references and a proof, see e.g. [14], Ch. VI.

2.2. Constructive worlds

An (infinite) *constructive world* is a countable set X (usually of some finite Bourbaki structures, such as the set of all error-correcting codes in a fixed alphabet, cf sec. 3.1 below) given together with a class of *structural numberings:* intuitively computable bijections $\nu : \mathbf{Z}_+ \to X$ which form a principal homogeneous space over the group of totally recursive permutations of $\mathbf{Z}_+$. A finite constructive world is any finite set.

Categorical Church's thesis, Part I. Let X, Y be two infinite constructive worlds, $\nu_X : \mathbf{Z}_+ \to X$ and $\nu_Y : \mathbf{Z}_+ \to X$ be their structural numberings, and F be an (intuitive) algorithm that takes as input an object $x \in X$ and produces an object $F(x) \in Y$ whenever x lies in the domain of definition of F; otherwise it outputs "NO" or works indefinitely.

Then $f := \nu_Y^{-1} \circ F \circ \nu_X : \mathbf{Z}_+ \to \mathbf{Z}_+$ is a partial recursive function.

Categorical Church's thesis, Part II. Let $\mathcal{C}$ be a category, whose objects are some infinite constructive worlds, and some finite constructive worlds of all finite cardinalities. Define the set of morphisms $\mathcal{C}(X, Y)$ to be the set partial maps that can be algorithmically computed.

Then $\mathcal{C}$ is equivalent to the category having one infinite object $\mathbf{Z}_+$, one finite object $\{1, \ldots, n\}$ of each cardinality, and partial recursive functions as morphisms. If X is finite, then $\mathcal{C}(X, Y)$ consists of all partial maps.

2.3. Kolmogorov complexity and Kolmogorov order

Let X be a constructive world. For any (semi-)computable function $u : \mathbf{Z}_+ \to X$, the (exponential) complexity of an object $x \in X$ relative to u is

$$K_u(x) := \min \{m \in \mathbf{Z}_+ \mid u(m) = x\}$$

If such m does not exist, we put $K_u(x) = \infty$.

Claim 2.1. *There exists such u ("an optimal Kolmogorov numbering", or "decompressor") that for each other (semi-)computable $v : \mathbf{Z}_+ \to X$, some constant $c_{u,v} > 0$, and all $x \in X$, one has*

$$K_u(x) \leq c_{u,v} K_v(x)$$

This $K_u(x)$ is called *Kolmogorov complexity* of x.

A Kolmogorov order of a constructive world X is a bijection $\mathbf{K} = \mathbf{K}_u : X \to \mathbf{Z}_+$ arranging elements of X in the increasing order of their complexities K_u.

Notice that any optimal numbering is only partial function, and its definition domain is not decidable. Moreover, the Kolmogorov complexity K_u itself is *not computable*: it is the lower bound of a sequence of computable functions.

The same can be said about the Kolmogorov order. Moreover, on $\mathbf{Z}_+$ it cardinally differs from the natural order in the following sense: it puts in the initial segments very large numbers that can be at the same time Kolmogorov simple. For example, let $a_n := n^{n^{\cdot^{\cdot^{\cdot^{n}}}}}$ (n times). Then $K_u(a_n) \leq cn$ for some $c > 0$.

In Sec. 4 below we will discuss other remarkable properties of complexity, in particular, its self-similar fractal properties.

Finally, the indeterminacy of the complexity related to different choices of optimal functions u, v is multiplicatively $\exp(O(1))$. The same is true for the Kolmogorov order.

For a thorough treatment of Kolmogorov complexity, cf [10]. Notice that in the literature one often uses the *logarithmic* Kolmogorov complexity which is defined as the length of the binary presentation of $K_u(x)$. It is interpreted as the length of the *maximally compressed description* of x. For our purposes, exponential version is more convenient, in particular, because it allows us to define an unambiguous Kolmogorov order on $\mathbf{Z}_+$ or any infinite constructive world.

2.4. Oracle assisted computations

The formal description of partial recursive functions in Sec. 2.1 (ii) allows one to define larger classes of partial functions that can be obtained by *oracle assisted* computations. The point is that the standard elementary operations can be applied to *arbitrary* partial functions. Therefore we can add any uncomputable (not partial recursive) functions to the list of basic functions and consider the minimal subset of partial functions containing this expanded list and closed wrt elementary operations.

This option was used in [20] in order to define the respective extensions of the notion of complexity and apply them to the explanation of Zipf's law in the situations, related to oracle assisted computations and library reuse, cf Sec. 4 below.

Formally, we are considering the (pro)perad generated by the elementary operations on partial functions and various algebras over it. It would be important to understand all relations between elementary operations. For the first steps in this direction, cf [26]; for a general formalism, cf [1].

3. Error-correcting codes and their asymptotic bounds

3.1. Basic notations

Choose an alphabet A, a finite set of cardinality $q \geq 2$. A *code* $C \subset A^n$ is a subset of words of length n. *Hamming distance* between two words of the same length is defined as

$$d((a_i), (b_i)) := \text{card}\{i \in (1, \dots, n) \, | \, a_i \neq b_i\}$$

Code parameters are the cardinality of the alphabet q and the numbers $n(C), k(C), d(C)$ defined by:

$$n(C) := n, \quad k(C) := k := [\log_q \text{card}(C)], \quad d(C) := d = \min\{d(a,b) \, | \, a, b \in C, a \neq b\}$$

Briefly, C is an $[n, k, d]_q$-code. Its *code point* is the point

$$x(C) := \left(\frac{k(C)}{n(C)}, \frac{d(C)}{n(C)}\right) \quad \in [0, 1]^2$$

Coordinates of $x(C) = (R(C), \delta(C))$ are called *transmission rate* and *relative distance*, respectively.

The idealized scheme of using error-correcting codes for information transmission can be described as follows. Some source data are encoded by a sequence of code words. After transmission through a noisy channel at the receiving end we will get a sequence of possibly corrupted words. If we know probability of corruption of a single letter, we can calculate, how many corrupted letters in a word we may allow for safe transmission; pairs of code words must be then separated by a larger Hamming distance. This necessity puts an upper bound on the achievable transmission rate.

A good code must maximize minimal relative distance when the transmission rate is chosen.

Our discussion up to now was restricted to *unstructured* codes: arbitrary subsets of words. Arguably, one more property of good codes is the existence of efficient algorithms of encoding and decoding. This can be achieved by introduction of *structured* codes. A typical choice is represented by *linear codes*: for them, A is a finite field of q elements, and C is a linear subspace of $\mathbf{F}_q^n$.

3.2. Asymptotic bound
Call *the multiplicity* of a code point the number of codes that project onto it.

Theorem 3.1 (Yu. Manin, 1981 and 2011)**.** *There exists a continuous function $\alpha_q(\delta)$, $\delta \in [0,1]$, with the following properties:*

(i) *The set of code points of infinite multiplicity is exactly the set of rational points $(R, \delta) \in [0,1]^2$ satisfying $R \le \alpha_q(\delta)$. The curve $R = \alpha_q(\delta)$ is called the asymptotic bound.*

(ii) *Code points x of finite multiplicity all lie strictly above the asymptotic bound and are called isolated ones: for each such point there is an open neighborhood containing x as the only code point.*

(iii) *The same statements are true for linear codes, with a possibly different asymptotic bound $R = \alpha_q^{lin}(\delta)$.*

3.3. Can one compute an asymptotic bound?
During the thirty years since the discovery of the asymptotic bounds, many upper and lower estimates were established for them, especially for the linear case: see the monograph [25]. Upper bounds helped to pinpoint a number of isolated codes.

However, the following most natural problems remain unsolved:

• To find an explicit formula for α_q or α_q^{lin}.

• To find any single value of $\alpha_q(\delta)$ or $\alpha_q^{lin}(\delta)$ for $0 < \delta < 1-q^{-1}$ (at the end segment $[1-q^{-1}, 1]$ these function vanish).

• To find any method of approximate computation of $\alpha_q(\delta)$ or $\alpha_q^{lin}(\delta)$.

• Clearly, $\alpha_q^{lin} \le \alpha_q$. Is this inequaliy *strict* somewhere?

3.4. A brief survey of some known results
(i) One can count the number of codes of bounded block length n and plot their code points. The standard probabilistic methods then give the following *Gilbert-Varshamov bounds*.

Most unstructured q-ary codes lie lower or only slightly above the Hamming curve

$$R = 1 - H_q(\delta/2), \quad H_q(\delta) = \delta \log_q(q-1) - \delta \log_q \delta - (1-\delta) \log_q(1-\delta)$$

Most linear q-ary codes lie near or only slightly above the Gilbert-Varshamov bound

$$R = 1 - H_q(\delta)$$

In particular,

$$\alpha_q(R) \ge \alpha_q^{lin}(R) \ge 1 - H_q(\delta)$$

(ii)   A useful combinatorial upper estimate is the Singleton bound:

$$R(C) + \delta(C) \leq 1 + 1/n(C)$$

Hence $\alpha_q(\delta) \leq 1 - \delta$. It follows that code points lying above this bound are isolated. The following Reed-Solomon (linear) codes $C \subset \mathbf{F}_q^n$ belong to this group.

Choose parameters $1 \leq k \leq n \leq q, d = n + 1 - k$. Choose pairwise distinct $x_1, \ldots, x_n \in \mathbf{F}_q$, Embed the space of polynomials $f(x) \in \mathbf{F}_q[x]$ of degree $\leq k - 1$ into $\mathbf{F}_q^n$ by

$$f \mapsto (f(x_1), \ldots, f(x_n)) \in \mathbf{F}_q^n$$

After works of Goppa, this construction was generalized. Points $x_1, \ldots, x_n \in \mathbf{F}_q$ were replaced by rational points of any smooth algebraic curve over $\mathbf{F}_q$, and polynomials by sections of an invertible sheaf. This allowed one to construct non-isolated linear codes lying partly strictly above the Gilbert-Varshamov bound.

This implies that we cannot "see" the asymptotic bound, plotting the set of (linear) code points of bounded size: we will see a cloud of points, whose upper bound concentrates near the Hamming or Varshamov-Gilbert bounds.

3.5. Partition function for codes involving complexity
The situation drastically changes, at least theoretically, if we allow ourselves to rearrange the codes in the order of growing Kolmogorov complexity.

In order to state our principal theorem, notice that the function $\alpha_q(\delta)$ is continuous and strictly decreasing for $\delta \in [1, 1 - q^{-1})$. Hence the limit points domain $R \leq \alpha_q(\delta)$ can be equally well described by the inequality $\delta \leq \beta_q(R)$ where β_q is the function inverse to α_q.

Fix an $R \in \mathbf{Q} \cap (0, 1)$. For $\Delta \in \mathbf{Q} \cap (0, 1)$, put

$$Z(R, \Delta; \beta) := \sum_{C: R(C) = R, \, \Delta \leq \delta(C) \leq 1} K_u(C)^{-\beta + \delta(C) - 1}$$

where K_u is an (exponential) Kolmogorov complexity on the constructive world of all codes in a given alphabet of cardinality q.

Theorem 3.2.   *(i) If $\Delta > \beta_q(R)$, then $Z(R, \Delta; \beta)$ is a real analytic function of β.*

(ii) If $\Delta < \beta_q(R)$, then $Z(R, \Delta; \beta)$ is a real analytic function of β for $\beta > \beta_q(R)$ such that its limit for $\beta - \beta_q(R) \to +0$ does not exist.

The following thermodynamical analogies justify our interpretation of the asymptotic bound a phase transition curve.

a) The argument β of the partition function corresponds to the inverse temperature.

b) The transmission rate R corresponds to the density ρ.

c) Our asymptotic bound transported into $(T = \beta^{-1}, R)$-plane as $T = \beta_q^{-1}(R)$ becomes the phase transition boundary in the (temperature, density)-plane.

4. Zipf's law and Kolmogorov order
4.1. Zipf's law
G. Zipf studied the frequencies with which words of a natural language are used in various texts. He found a remarkably stable pattern [27, 28]: if all words w_k of a language are ranked according to decreasing frequency of their appearance in a representative corpus of texts, then the frequency p_k of w_k is approximately inversely proportional to its rank k: see e.g. Fig. 1 in [12] based upon a corpus containing $4 \cdot 10^7$ Russian words.

Zipf himself has suggested that this distribution "minimizes effort". Mandelbrot in [11] has shown that if we postulate and denote by C_k a certain "cost" (of producing, using etc.) of the word of rank k, then the frequency distribution $p_k \sim 2^{-h^{-1}C_k}$ minimizes the ratio $h = C/H$, where $C := \sum_k p_k C_k$ is the average cost per word, and $H := -\sum_k p_k \log_2 p_k$ is the average entropy, see [13].

We get from this a power law, if $C_k \sim \log k$. An additional problem, what is so special about power -1, must be addressed separately.

In all such discussions, it is more or less implicitly assumed that empirically observed distributions concern fragments of a potential countable infinity of objects. In the mathematical model suggested in [20] it is assumed that these objects form an infinite constructive world in the sense of 2.2 above. Below I will survey this model.

4.2. How minimization of complexity leads to Zipf's law
A mathematical model of Zipf's law is based upon two postulates:

(A) *Rank ordering coincides with a Kolmogorov ordering (up to a factor $exp\,(O(1))$), cf 2.3 above.*

(B) *The probability distribution producing Zipf's law (with exponent -1) is (an approximation to) the L. Levin maximal computable from below distribution: see [29, 8, 9, 10].*

If we accept (A) and (B), then Zipf's law follows from two basic properties of Kolmogorov complexity:

(a) *rank of w defined according to (A) is $exp\,(O(1)) \cdot K(w)$.*

(b) *Levin's distribution assigns to an object w probability $\sim KP(w)^{-1}$ where KP is the exponentiated prefix Kolmogorov complexity (cf [10, 2]), and we have, up to $exp(O(1))$-factors,*

$$K(w) \preceq KP(w) \preceq K(w) \cdot \log^{1+\varepsilon} K(w)$$

with arbitrary $\varepsilon > 0$.

There is a slight discrepancy between the growth orders of K and KP. This discrepancy ensures the convergence of the series $\sum_w KP(w)^{-1}$. On finite sets of data this small discrepancy is additionally masked by the dependence of both K and KP on the choice of an optimal encoding.

"Minimization of effort" is thus achieved if effort itself is interpreted as the length of the maximally compressed prefix free description of an object.

Such a picture makes sense especially if the objects satisfying Zipf's distribution, are *generated* rather than simply *observed.*

This matches very well the results of the previous section on asymptotic bounds for error-correcting codes: if one produces codes in the order of their Kolmogorov complexity rather than size, their code points will well approximate the picture of the whole domain under the asymptotic bound. Moreover, Levin's distribution very naturally leads to the thermodynamic partition function on the set of codes, and to the interpretation of asymptotic bound as a phase transition curve. In sec. 2, we have written it in the form the form $\sum_C K(C)^{-s(C)}$ where $s(C)$ is a certain function defined on codes and including as parameters analogs of temperature and density. We could replace K with KP, and freely choose the optimal family defining complexity: this would have no influence at all on the form of the phase curve/asymptotic bound.

It is interesting to observe that the mathematical problem of generating good error-correcting codes historically made a great progress in the 1980's with the discovery of algebraic geometric

Goppa codes, that is precisely with the discovery of greatly compressed descriptions of large combinatorial objects.

To summarize, the class of a priori probability distributions that we are considering here is *qualitatively distinct* from those that form now a common stock of sociological and sometimes scientific analysis: cf a beautiful synopsis of the latter by Terence Tao in [23] who also stresses that "mathematicians do not have a fully satisfactory and convincing explanation for how the Zipf's law comes about and why it is universal".

What arguments could furnish such an explanation? Ubiquity of Gaussian distribution, for example, is often explained away by appealing to the central limit theorem: average of many independent random (equally distributed) variables tends to be Gaussian for whatever initial distribution. Below I will argue that universality of Zipf's law is similarly based on the surprisingly self-similar nature of Kolmogorov complexity.

4.3. Fractal landscape and self-similarity of the Kolmogorov complexity

In [10] (pp 103, 105, 178), one can find a schematic graph of logarithmic complexity of naturals. The visible "continuity" of this graph reflects the fact that complexity of $k+1$ in any reasonable encoding is almost the same as complexity of k. It looks as follows: most of the time it follows closely the graph of $\log k$, but infinitely often it drops down, lower than any given computable function:

One does not see or suspect self-similarity. But it is there: if one restricts this graph onto any infinite decidable subset of $\mathbf{Z}_+$ in increasing order, one will get the same complexity relief as for the whole $\mathbf{Z}_+$: in fact, for any recursive bijection f of $\mathbf{Z}_+$ with a subset of $\mathbf{Z}_+$ we have $K(f(x)) = \exp(O(1)) \cdot K(x)$.

If we pass from complexity to a Levin's distribution, that is, basically, invert the values of complexity, these fractal properties survive.

This property can be read as the extreme stability of such a distribution with respect to the passage to various sub-universes of objects, computable renumbering of objects etc., in the same way as the picture of random noise in a stable background is held responsible for universality of normal distribution.

4.4. Complexity on the background of oracle assisted computations and library reuse

In [24] and [arXiv:cs/0508023] T. Veldhuizen considers Zipf's law in an unusual context that did not exist in the days when Kolmogorov, Solomonov and Chaitin made their ground-breaking discoveries, but which provides, in a sense, landscape for an industrial incarnation of complexity. Namely, Veldhuizen studies actual software and software libraries and analyzes possible profits from software reuse. Metaphorically, this is a picture of human culture whose everyday existence depends on a continuous reuse of treasures created by researchers, poets, philosophers, cf [Manin Y I arXiv:1301.0081].

Mathematically, reuse furnishes new tools of compression: roughly speaking, a function f may have a very large Kolmogorov complexity, but the length of the library address of its program may be short, and only the latter counts if one can simply copy the program from the library.

In order to create a mathematical model of reuse and its Zipf's landscape, the notion of an admissible set of partial functions note, I need to define the mathematical notion of *relative Kolmogorov complexity $K(f|\Phi)$.*

4.5. Admissible sets of functions

Consider a set Φ of partial functions $f : (\mathbf{Z}_+^+)^m \to (\mathbf{Z}_+^+)^n$, $m, n \geq 0$. We will call Φ *an admissible set*, if it is countable and satisfies the following conditions.

(i) Φ is closed under composition and contains all projections (forget some coordinates), and embeddings (permute and/or add some constant coordinates).
Any $(m+1, n)$-function can be considered as a family of (m, n)-functions

$$(u_k) : u_k(x_1, \ldots, x_m) := u(x_1, \ldots, x_m, k)$$

From (i) it follows that for any $u \in \Phi$ and $k \in \mathbf{Z}_+^+$, also $u_k \in \Phi$. Similarly, if $u(x_1, \ldots, x_m)$ is in Φ, then

$$U(x_1, \ldots, x_m, x_{m+1}, \ldots, x_{m+n}) \equiv u(x_1, \ldots, x_m)$$

is in Φ.

(ii) For any (m, n), there exists such an $(m+1, n)$-function $u \in \Phi$ that the family of functions $u_k : (\mathbf{Z}_+^+)^m \to (\mathbf{Z}_+^+)^n$, contains all (m, n)-functions belonging to Φ. We will say that such a function u (or family (u_k)) is ample.

(iii) Let f be a total recursive function f whose image is decidable, and f defines a bijection between $D(f)$ and image of f. Then Φ contains both f and f^{-1}.

It is shown in [20] that one can define analog of complexity with respect to such a set, $K(x \mid \Phi)$ and, moreover, that such sets can be obtained as "algebras" over a (pro)perad generated by standard operations that usually are applied only to partially recursive functions.

There are many instances of empiric Zipf's laws where our picture might be applicable: cf [6, 5, 4] and also [Murtra B C and Solé R 2010 *On the Universality of Zipf's Law*, Santa Fe Institute, unpublished, available online]. Such a reduction of the Zipf law for natural languages might require for its justification some neurobiological data: cf [12], appendix A in [arXiv:0710.0105].

5. Feynman graphs and perturbation series in quantum physics
5.1. A toy model
Feynman path integral is an heuristic expression of the form

$$\frac{\int_{\mathcal{P}} e^{S(\varphi)} D(\varphi)}{\int_{\mathcal{P}} e^{S_0(\varphi)} D(\varphi)} \tag{5.1}$$

or, more generally, a similar heuristic expression for *correlation functions*.

In the expression (5.1), $\mathcal{P}$ is imagined as a functional space of *classical fields* φ on a *space-time manifold* M; $S : \mathcal{P} \to \mathbf{C}$ is a functional of *classical action* measured in Planck's units. S_0 is its *quadratic part*, or *"free field action"*.

Usually $S(\varphi)$ itself is an integral over M of a local density on M called *Lagrangian*. In our notation

$$S(\varphi) = -\int_M L(\varphi(x)) dx$$

The Lagrangian density may depend on derivatives, include distributions etc.

Finally, the integration measure $D(\varphi)$ and the integral itself $\int_{\mathcal{P}}$ should be considered as symbolic constituents of the total expression (4.1) conveying a vague but powerful idea of *"summing quantum amplitudes over virtual classical trajectories"*.

In our toy model, we will replace $\mathcal{P}$ by a finite-dimensional real space. We endow it with a basis indexed by a finite set of "colors" A, and an Euclidean metric g encoded by the symmetric tensor $(g^{ab})_{a,b \in A}$. We put $(g^{ab}) = (g_{ab})^{-1}$.

The action functional $S(\varphi)$ is a formal series in linear coordinates (φ^a) on $\mathcal{P}$, of the form

$$S(\varphi) = S_0(\varphi) + S_1(\varphi)$$

where

$$S_0(\varphi) := -\frac{1}{2}\sum_{a,b} g_{ab}\varphi^a\varphi^b \quad \text{and} \quad S_1(\varphi) := \sum_{k=1}^{\infty}\frac{1}{k!}\sum_{a_1,\dots,a_k\in A} C_{a_1,\dots,a_k}\varphi^{a_1}\dots\varphi^{a_k} \tag{5.2}$$

where $(C_{a_1,\dots,a_n})$ are certain symmetric tensors.

Below we will consider (g_{ab}) and $(C_{a_1,\dots,a_n})$ as independent formal variables, *"formal coordinates on the space of theories"*.

We will express the toy version of (5.1) as a formal series over (isomorphism classes of) graphs.

A (combinatorial) graph τ, by definition, consists of two finite sets: flags F_τ and vertices V_τ. Besides, an involution j_τ of F_τ is given, showing which pairs of flags form halves of edges, and which are not (tails). Finally, the map $\partial_\tau : F_\tau \to V_\tau$ shows to which vertex each graph is incident. The geometric realization of τ is a topological space whose structure is suggested by the choice of words in the definition given by Figure 1:
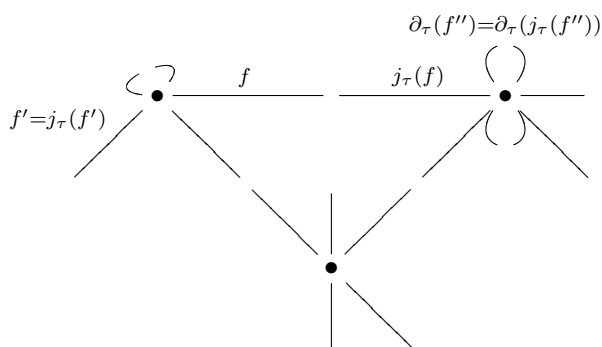

Figure 1. Geometric realization of τ

Each edge e consists of a pair of flags denoted ∂e, and each vertex v determines the set of flags incident to it denoted $F_\tau(v)$. By $\chi(\tau)$ we denote the Euler characteristic of the geometric realization of τ.

Theorem 5.1. *Let λ be a formal parameter. Then*

$$\frac{\int_{\mathcal{P}} e^{\lambda^{-1}S(\varphi)}D(\varphi)}{\int_{\mathcal{P}} e^{\lambda^{-1}S_0(\varphi)}D(\varphi)} = \sum_{\tau\in\Gamma}\frac{\lambda^{-\chi(\tau)}}{|\mathrm{Aut}\,\tau|}\,w(\tau) \tag{5.3}$$

where τ runs over isomorphism classes of all finite graphs τ. The weight $w(\tau)$ of such a graph is determined by the action functional (1.2) as follows:

$$w(\tau) := \sum_{u:F_\tau\to A}\prod_{e\in E_\tau} g^{u(\partial e)}\prod_{v\in V_\tau} C_{u(F_\tau(v))} \tag{5.4}$$

More precisely, the identity (5.2) is obtained by first interpreting the integrands in the numerator of (5.2) as formal series in $(g_{ab}, C_{a_1,\dots,a_k})$, and then integrating term-wise by using the well known formulas for Gaussian integrals.

6. Graphs as flowcharts, and Hopf algebras

6.1. Graphs as flowcharts

Feynman diagrams of more realistic models and graphs used in the computation theory can be considered as *flowcharts* describing the flow of information from a part of tails playing role of *inputs* to another part, playing role of *outputs*. At vertices, the information gets processed.

In order to make such an interpretation workable, we need pay more attention to orientation. *Orientation* of a graph τ is the decoration $F_\tau \to L_F = \{in, out\}$ such that halves of any edge are decorated by different labels.

Tails of τ oriented *in* (resp. *out*) are called *(global) inputs* T_τ^{in} (resp. *(global) outputs* T_τ^{out}) of τ. Similarly, $F_\tau(v)$ is partitioned into inputs and outputs of the vertex v.

An oriented graph τ is called *directed* if it satisfies the following condition:

On each connected component one can define a continuous real valued function ("time") in such a way that moving in the direction of orientation along each flag inreases the value of this function.

In particular, oriented trees and forests are always directed, and physical Feynman diagrams without loops as well.

An abstract *flowchart* is a directed graph endowed with the decoration of its vertices by a set Op of (names of) operations that can be performed on certain inputs producing certain outputs. Generally, flags are also labeled by *types* of the arguments.

To be more precise, flowcharts in theoretical computer science form a natural hierarchy.

At the lower level of this hierarchy, *histories of computations* are situated. For example, the sequence of the states of a Turing machine, performing a concrete computation, may be encoded by a flowchart, in which inputs of all vertices are decorated by 0 or 1, and vertices themselves carry either the name of identical operation or the name of the internal state of the head, reading the respective site. Such a history may well be *infinite*.

At higher levels flowcharts may serve as *descriptions*: programs represented as compositions of some subprograms, but not specifying concrete values of arguments and thus hiding the actual computation process and/or compressing the notation.

We omit here a formal definition of admissible sets of decorated flowcharts: cf [17, 18] for further details. Briefly, an admissible set must be closed wrt finite disjoint unions and *cuts* that will be defined below.

For another version of flowcharts see [22].

6.2. Connes-Kreimer bialgebras of flowcharts [3]

Let Fl be an admissible set of decorated graphs, $k :=$ a commutative ring. We denote by $H = H_{Fl}$ the k-linear span of isomorphism classes $[\tau]$ of graphs τ in Fl and define multiplication by

$$m : H \otimes H \to H, \quad m\left([\sigma] \otimes [\tau]\right) := \left[\sigma \coprod \tau\right]$$

We pass now to cuts and comultiplication.

Let τ be an oriented graph. Call *a proper cut C* of τ any partition of V_τ into a disjoint union of two non-empty subsets V_τ^C (upper vertices) and $V_{\tau,C}$ (lower vertices) satisfying the following conditions:

(i) For each oriented wheel in τ, all its vertices belong either to V_τ^C, or to $V_{\tau,C}$.

(ii) If an edge e connects a vertex $v_1 \in V_\tau^C$ to $v_2 \in V_{\tau,C}$, then it is oriented from v_1 to v_2 ("information flows only from past to future").

(iii) Two improper cuts: $\tau^C := \tau$ or $\tau_C = \tau$.

Denote by τ^C (resp. τ_C) the subgraphs of τ consisting of vertices V_τ^C (resp. $V_{\tau,C}$) and incident flags. Put

$$\Delta : H \to H \otimes H, \quad \Delta([\tau]) := \sum_C \left[\tau^C\right] \otimes [\tau_C]$$

sum being taken over all cuts of τ.

Claim 6.1. *(i) m defines on H the structure of a commutative k-algebra with unit $[\emptyset]$. Set $\eta : k \to H$ and $1_k \mapsto [\emptyset]$.*

(ii) Δ is a coassociative comultiplication on H with counit

$$\varepsilon : H \to k, \quad \sum_{\tau \in Fl} a_{[\tau]}[\tau] \mapsto a_{[\emptyset]}$$

(iii) $(H, m, \Delta, \varepsilon, \eta)$ is a commutative bialgebra with unit and counit.

Theorem 6.2 (K. Ebrahimi-Fard, D. Manchon [7]). *H is a Hopf algebra (i. e. has a unique antipode) if one can introduce an grading on H such that*

$$m(H_p \otimes H_q) \subset H_{p+q}, \quad \Delta(H_n) \subset \oplus_{p+q=n} H_p \otimes H_q$$

and moreover, $H_0 = k[\emptyset]$ is one-dimensional, so that H is connected.

A possible choice of such grading:

$$H_n := \text{the } k\text{-submodule of } H \text{ spanned by } [\tau] \in Fl \text{ with } |F_\tau| = n.$$

7. Regularization and renormalization
7.1. Regularization by "minimal subtraction"
Generally, by regularization we mean "producing a finite answer from infinite one". A typical example is this.

Consider the ring $\mathcal{A}$ of germs of meromorphic functions of z at $z = 0$. Put $\mathcal{A}_- := z^{-1}\mathbf{C}[z^{-1}]$, and denote by $\mathcal{A}_+$ the ring of germs of regular functions at $z = 0$. The value of regular function at zero is $\varepsilon_{\mathcal{A}}(f) := f(0)$. Any germ is unique sum of regular one and one belonging to $\mathcal{A}_-$.

If a function is not necessarily regular, the regularized value of f at 0 is $\varepsilon_{\mathcal{A}}(f_+) = f_+(0)$, where

$$f_+(z) := f(z) - \text{the polar part of } f.$$

Generally, a "minimal subtraction algebra" is a commutative associative K-algebra $\mathcal{A}$ represented as the direct sum of two linear subspaces $\mathcal{A} = \mathcal{A}_+ \oplus \mathcal{A}_-$, each being a subalgebra. Usually $\mathcal{A}$ is unital and $1 \in \mathcal{A}$; besides, we have an augmentation homomorphism $\varepsilon_{\mathcal{A}} : \mathcal{A}_+ \to K$.

7.2. Connes-Kreimer renormalization
This is a version of regularization that:

(i) is performed simultaneously for an infinite family of functions indexed by flowcharts;

(ii) uses the "division by the collective pole part" in a noncommutative group in place of subtraction of an individual pole.

More precisely, consider a Hopf K-algebra $\mathcal{H}$, and a minimal subtraction unital algebra $\mathcal{A}_+, \mathcal{A}_- \subset \mathcal{A}, \varepsilon_{\mathcal{A}} : \mathcal{A} \to K$.

Denote by $G(\mathcal{A})$ the group of K-linear maps $\varphi : \mathcal{H} \to \mathcal{A}$ such that $\varphi(1_{\mathcal{H}}) = 1_{\mathcal{A}}$, with the convolution product

$$\varphi * \psi(x) := m_{\mathcal{A}}(\varphi \otimes \psi)\Delta(x) = \varphi(x) + \psi(x) + \sum_{(x)} \varphi(x')\psi(x'')'$$

identity $e(x) := u_{\mathcal{A}} \circ \varepsilon(x)$, and inversion

$$\varphi^{*-1}(x) = e(x) + \sum_{m=1}^{\infty} (e - \varphi)^{*m}(x)$$

In situations that we will consider, for any $x \in \ker \varepsilon$ the latter sum contains only finitely many non-zero summands.

We will say that φ is *a character* if it is a homomorphism of algebras.

Following Birkhoff, we may now define "collective pole" and "collective regular part" of φ. More precisely, if $\mathcal{A}$ is a minimal subtraction algebra, each $\varphi \in G(\mathcal{A})$ admits a unique decomposition of the form

$$\varphi = \varphi_-^{*-1} * \varphi_+, \quad \varphi_-(1) = 1_{\mathcal{A}}, \quad \varphi_-(\ker \varepsilon) \subset \mathcal{A}_-, \quad \varphi_+(\mathcal{H}) \subset \mathcal{A}_+$$

Values of renormalized polar (resp. regular) parts φ_- (resp. φ_+) on $\ker \varepsilon$ are given by the inductive formulas

$$\varphi_-(x) = -\pi\left(\varphi(x) + \sum_{(x)} \varphi_-(x')\varphi(x'')\right), \quad \varphi_+(x) = (\mathrm{id} - \pi)\left(\varphi(x) + \sum_{(x)} \varphi_-(x')\varphi(x'')\right)$$

Here $\pi : \mathcal{A} \to \mathcal{A}_-$ is the polar part projection in the algebra $\mathcal{A}$.

Physicists invented these inductive formulas: they are known as BPZH-renormalization, for Bogolyubov-Parasyuk-Zimmermann-Hepp.

7.3. Deforming the Halting Problem

Let f be a partial recursive function. The Halting Problem for f is that of recognizing whether a number $k \in \mathbf{Z}_+$ belongs to its definition domain $D(f)$. In this subsection, we will translate it into the problem, whether an analytic function $\Phi(k, f; z)$ of a complex parameter z has a pole at $z = 1$.

The relevant minimal subtraction algebra will be a version of our example from 6.1. Let $\mathcal{A}_+$ be the algebra of analytic functions in $|z| < 1$, continuous at $|z| = 1$, $\varepsilon_{\mathcal{A}} : \Phi(z) \mapsto \Phi(1)$. Put

$$\mathcal{A}_- := (1 - z)^{-1}\mathbf{C}[(1 - z)^{-1}], \quad \mathcal{A} := \mathcal{A}_+ \oplus \mathcal{A}_-$$

We now choose an appropriate programming method P and construct its Hopf algebra. Basically, $\mathcal{H} = \mathcal{H}_P$ is the symmetric algebra, spanned by isomorphism classes $[p]$ of certain descriptions. Comultiplication in $\mathcal{H}_P$ is dual to the composition of descriptions.

The main choice is that of characters, corresponding to the halting problem.

The character $\varphi_k : \mathcal{H}_P \to \mathcal{A}$ corresponding to the halting problem at a point $k \in \mathbf{Z}_+$ for the partial recursive function computable with the help of a description $p \in P(\mathbf{Z}_+, \mathbf{Z}_+)$, will be defined as $\varphi_k([p]) := \Phi(k, f; z) \in \mathcal{A}$ where the function Φ is described below.

Using the trick used in the theory of quantum computation (usually applied in the context of finite automata) we will first reduce the general halting problem to the recognition of fixed points of permutations.

Start with a partial recursive function $f : X \to X$, where X is a constructive world. Extend X by one point, i. e. form $X \coprod \{*_X\}$. Choose a total recursive structure of an additive group without torsion on $X \coprod \{*_X\}$ with zero $*_X$. Extend f to the everywhere defined function

$$g : X \coprod \{*_X\} \to X \coprod \{*_X\}, \quad g(y) := *_X \text{ if } y \notin D(f)$$

Define

$$\tau_f : \left(X \coprod \{*_X\} \right)^2 \to \left(X \coprod \{*_X\} \right)^2, \quad \tau_f(x, y) := (x + g(y), y)$$

It is a permutation. Since $(X \coprod \{*_X\}, +)$ has no torsion, the only finite orbits of $\tau_f^{\mathbf{Z}}$ are fixed points.

Moreover, the restriction of τ_f upon the recursive enumerable subset $D(\sigma_f) := (X \coprod \{*_X\}) \times D(f)$ induces a partial recursive permutation σ_f of this subset. Since $g(y)$ never takes the zero value $*_X$ on $y \in D(f)$, but always is zero outside it, the complement to $D(\sigma_f)$ in Y consists entirely of fixed points of τ_f. Thus, the halting problem for f reduces to the fixed point recognition for τ_f.

7.4. The halting problem renormalization character

Define a Kolmogorov numbering on a constructive world X as a bijection $\mathbf{K} = \mathbf{K}_u : X \to \mathbf{Z}_+$ arranging elements of X in the increasing order of their complexities K_u.

Let $\sigma : X \to X$ be a partial recursive map, such that σ maps $D(\sigma)$ to $D(\sigma)$ and induces a permutation of this set. Put $\sigma_{\mathbf{K}} := \mathbf{K} \circ \sigma \circ \mathbf{K}^{-1}$ and consider this as a permutation of the subset

$$D(\sigma_{\mathbf{K}}) := \mathbf{K}(D(\sigma)) \quad \subset \mathbf{Z}_+$$

consisting of numbers of elements of $D(\sigma)$ in the Kolmogorov order.

If $x \in D(\sigma)$ and if the orbit $\sigma^{\mathbf{Z}}(x)$ is infinite, then there exist such constants $c_1, c_2 > 0$ that for $k := \mathbf{K}(x)$ and all $n \in \mathbf{Z}$ we have

$$c_1 \cdot \mathbf{K}(n) \leq \sigma_{\mathbf{K}}^n(k) \leq c_2 \cdot \mathbf{K}(n)$$

Now let $X = \mathbf{Z}_+$ and let σ be a partial recursive map, inducing a permutation on its definition domain. Put

$$\Phi(k, \sigma; z) := \frac{1}{k^2} + \sum_{n=1}^{\infty} \frac{z^{\mathbf{K}(n)}}{\left(\sigma_{\mathbf{K}}^n(k) \right)^2}$$

Then we have:

Theorem 7.1. (i) *If σ-orbit of x is finite, then $\Phi(x, \sigma; z)$ is a rational function in z whose all poles are of the first order and lie at roots of unity.*

(ii) *If this orbit is infinite, then $\Phi(x, \sigma; z)$ is the Taylor series of a function analytic at $|z| < 1$ and continuous at the boundary $|z| = 1$.*

References

[1] Borisov D and Manin Y 2008 Generalized operads and their inner cohomomorhisms In *Geometry and Dynamics of Groups and Spaces* (In memory of Aleksander Reznikov) eds M Kapranov *et al* Progress in Math **265** (Birkhäuser) p 247

[2] Calude Ch S and Staiger L 2009 *Math. Struct. Comput. Sci.* **19** 45

[3] Connes A and Kreimer D 2000 *Comm. Math. Phys.* **210** 249

[4] Dehaene S 1997 *The Number Sense. How the Mind Creates Mathematics* (Oxford Univ. Press)

[5] Dehaene S and Mehler J 1992 *Cognition* **43** 1

[6] Delahaye J-P 2012 *Pour la Science* **421** 80

[7] Ebrahimi-Fard K and Manchon D 2009 *IRMA Lect. Math. Theor. Phys.* **15** 179 (Zürich: EMS)

[8] Levin L A 1976 *Soviet Math. Dokl.* **17** 522

[9] Levin L A 1984 *Information and Control* **61** 15

[10] Ming Li and Vitányi P 1993 *An Introduction to Kolmogorov Complexity and its Applications* (Springer-Verlag)

[11] Mandelbrot B 1953 An Information Theory of the Statistical Structure of Languages ed W Jackson *Communication Theory* (Woburn, MA: Butterworth) p 486

[12] Manin D Yu 2008 *Cognitive Science* **32** 1075

[13] Manin D Yu 2009 *J. Quantitative Linguistics* **16** 274

[14] Manin Yu I 2010 *A Course in Mathematical Logic for Mathematicians* 2nd edition Graduate Texts in Math. **53** (Springer-Verlag)

[15] Manin Yu 2012 *WTCS 2012 (Calude Festschrift)* **LNCS 7160** p 174

[16] Manin Yu 1999 Classical computing, quantum computing, and Shor's factoring algorithm *Séminaire Bourbaki* **862** (June 1999); 2000 *Astérisque* **266** 375

[17] Manin Yu 2012 Renormalization and computation I. Motivation and background eds J-L Loday and B Vallette *Proceedings OPERADS 2009 Séminaires et Congrès 26* Soc Math de France p 181

[18] Manin Yu 2012 *Math. Struct. Comp. Sci.* **22** Special issue (Cambridge Univ. Press) 729

[19] Manin Yu 2013 Kolmogorov complexity as a hidden factor of scientific discourse: from Newton's law to data mining Talk at the Plenary Session of the Pontifical Academy of Sciences on *Complexity and Analogy in Science: Theoretical, Methodological and Epistemological Aspects* Vatican 5-7 November 2012

[20] Manin Yu 2014 *Funct. Anal. Appl.* **48** 116

[21] Manin Yu and Marcolli M 2014 *J. Differential Geom.* **97** 91

[22] Scott D 1971 The lattice of flow diagrams *Symposium on Semantics of Algorithmic Languages* Lecture Notes in Math. **188** p 311

[23] Tao T 2012 *Daedalus* **141** 23

[24] Veldhuizen T L 2005 Software libraries and their reuse: entropy, Kolmogorov complexity, and Zipf's law *Library-Centric Software Design* (LCSD 2005) an OOPSLA 2005 workshop

[25] Vladut S G, Nogin D Yu and Tsfasman M A 2007 *Algebraic Geometric Codes: Basic Notions* Mathematical Surveys and Monographs **139** (Providence, RI: AMS)

[26] Yanofsky N S 2011 *J. Logic Comput.* **21** 253

[27] Zipf G K 1936 *The Psycho-Biology of Language* (London: Routledge)

[28] Zipf G K 1949 *Human Behavior and the Principle of Least Effort* (Addison-Wesley)

[29] Zvonkin A K and Levin L A 1970 *Uspehi Mat. Nauk* **25** 8