INFINITIES IN QUANTUM FIELD THEORY AND IN CLASSICAL COMPUTING: RENORMALIZATION PROGRAM

Yuri I. Manin

PLAN

INTRODUCTION

• The main observable quantities in Quantum Field Theory, *correlation functions*, **are expressed by Feynman path integrals. A mathematical definition of them involving a measure and actual integration is still lacking. Instead, it is replaced by a series of** *ad hoc* **but highly efficient and suggestive heuristic formulas such as** *perturbation formalism*.

• Perturbation formalism interprets such an integral as a formal series of finite–dimensional but *divergent integrals,* **indexed by Feynman graphs.**

• *Renormalization* **is a prescription that allows one to systematically "subtract infinities" from these divergent terms producing an asymptotic series for quantum correlation functions.**

INTRODUCTION

• The main observable quantities in Quantum Field Theory, *correlation functions*, **are expressed by Feynman path integrals. A mathematical definition of them involving a measure and actual integration is still lacking. Instead, it is replaced by a series of** *ad hoc* **but highly efficient and suggestive heuristic formulas such as** *perturbation formalism*.

• Perturbation formalism interprets such an integral as a formal series of finite–dimensional but *divergent integrals,* **indexed by Feynman graphs.**

• *Renormalization* **is a prescription that allows one to systematically "subtract infinities" from these divergent terms producing an asymptotic series for quantum correlation functions.**

• Graphs treated as *"flowcharts"***, also form a combinatorial skeleton of the abstract computation theory.**

• Infinities in Computation Theory arise from infinite loops/searches in infinite haystacks for a needle which is not there.

• In this paper I argue that such infinities can be addressed in the same way as Feynman divergences.

1. FEYNMAN GRAPHS AND PERTURBATION SERIES: A TOY MODEL

• *Feynman path integral* **is an heuristic expression of the form**

$$\frac{\int_{\mathcal{P}} e^{S(\varphi)} D(\varphi)}{\int_{\mathcal{P}} e^{S_0(\varphi)} D(\varphi)} \tag{1.1}$$

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

• In the expression (1.1), $\mathcal{P}$ **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.**

• Usually $S(\varphi)$ **itself is an integral over** M **of a local density on** M **called** *Lagrangian.* **In our notation** $S(\varphi) = -\int_M L(\varphi(x))dx$. **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 (1.1) conveying a vague but powerful idea of "summing over 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 on $\mathcal{P}$, (φ^a), of the form

$$S(\varphi) = S_0(\varphi) + S_1(\varphi), \quad S_0(\varphi) := -\frac{1}{2}\sum_{a,b} g_{ab}\varphi^a\varphi^b,$$

$$S_1(\varphi) := \sum_{k=1}^{\infty} \frac{1}{k!} \sum_{a_1,\dots,a_k \in A} C_{a_1,\dots,a_k}\varphi^{a_1}\dots\varphi^{a_k} \tag{1.2}$$

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 (1.1) as a series over (isomorphism classes of) graphs.

• A graph τ for us consists of two finite sets, edges E_τ and vertices V_τ, and the incidence map sending E_τ to the set of unordered pairs of vertices. Halves of edges form flags F_τ.

• THEOREM. We have, for a formal parameter λ

$$\frac{\int_{\mathcal{P}} e^{\lambda^{-1}S(\varphi)}D(\varphi)}{\int_{\mathcal{P}} e^{\lambda^{-1}S_0(\varphi)}D(\varphi)} = \sum_{\tau\in\Gamma} \frac{\lambda^{-\chi(\tau)}}{|\mathrm{Aut}\,\tau|}\, w(\tau) \tag{1.3}$$

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:

$$w(\tau) := \sum_{u\colon F_\tau\to A} \prod_{e\in E_\tau} g^{u(\partial e)} \prod_{v\in V_\tau} C_{u(F_\tau(v))}\,. \tag{1.4}$$

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

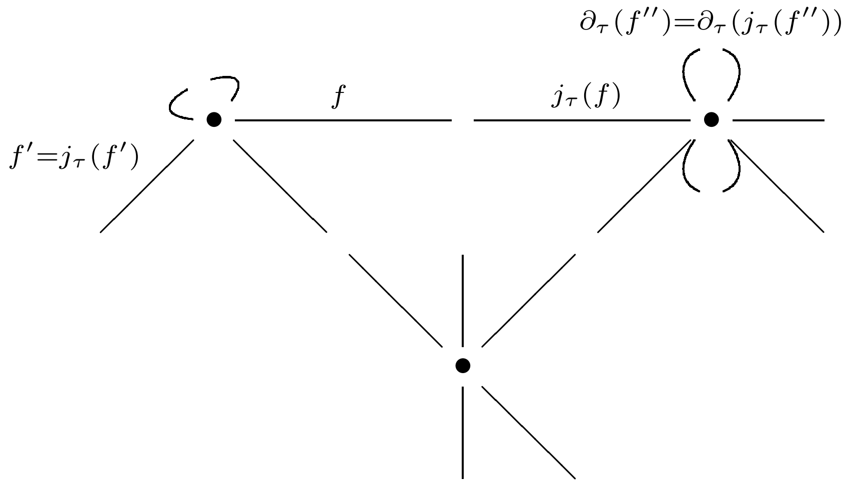
Some explanations:

- *Making sense of the equality (1.3) in the toy model:*

(i) Replace the exponent in the numerator integrand by $e^{\lambda^{-1}S_0(\varphi)}$ **times the formal series in** $C_{a_1,\ldots,a_k}, \lambda^{-1}$**.**

(ii) Formally integrate the series termwise interpreting the exponential term as Gaussian integral. Use Wick's Lemma to this end.

- *Visualizing graphs:*



- *How weights $w(\tau)$ are produced from decorated graphs:*

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

One decoration = a map $u : F_\tau \to A$, "coloring flags".

It induces decorations of edges by variables $g^{u(\partial e)}$ and decorations of vertices by variables $C_{u(F_\tau(v))}$

A fixed decoration produces a monomial in formal coordinates; the whole sum is a sum over isomorphism classes of decorated graphs.

• *How weights $w(\tau)$ are produced from decorated graphs:*

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

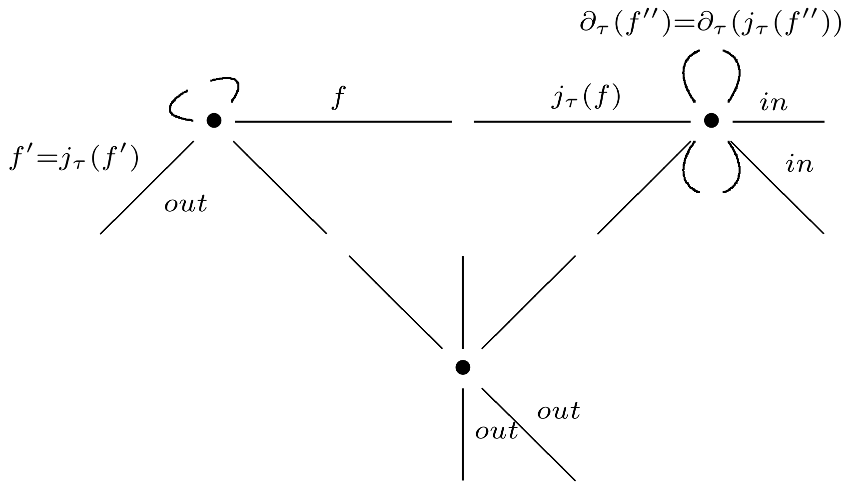
One decoration = a map $u:\ F_\tau \to A$, "coloring flags".

It induces decorations of edges by variables $g^{u(\partial e)}$ and decorations of vertices by variables $C_{u(F_\tau(v))}$

A fixed decoration produces a monomial in formal coordinates; the whole sum is a sum over isomorphism classes of decorated graphs.

• NB The graphs occurring in toy model are not oriented and have no "free flags" = "tails", "leaves" etc.

In the applications to computation, they will certainly have tails symbolizing inputs and outputs, and will be "time"– oriented.



∂τ(f″)=∂τ(jτ(f″))
f
jτ(f)
in
in
f′=jτ(f′)
out
out
out

2. GRAPHS, FLOWCHARTS, AND HOPF ALGEBRAS

• In order not to mix illustrative pictures with mathematical structures, we will distinguish between *combinatorial graphs* **and their** *geometric realizations***.**

• A combinatorial flag is a family of sets and maps:

$$\tau := (F_\tau, V_\tau, \partial_\tau : F_\tau \to V_\tau, j_\tau : F_\tau \to F_\tau), \quad j_\tau^2 = id.$$

Combinatorial graphs form a *category***, with various classes of (not at all obvious) morphisms, serving different purposes in different contexts. Only** *isomorphisms* **are evident.**

• Geometric realization of τ is a topological space glued from segments $[0, 1/2]$ indexed by F_τ: first use ∂_τ to produce corollas of vertices by gluing 0's,, then use j_τ to collect together corollas by gluing $1/2$'s.

• Let $L = (L_F, L_V)$ be two sets: *labels of flags and vertices***, respectively.**

An L–decoration **of the combinatorial graph τ consists of two maps $F_\tau \to L_F$, $V_\tau \to L_V$. Usually these maps are restricted by certain** *compatibility with incidence relations***.**

• Orientation := 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 of the geometric realization $|\tau|$, 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.

• 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.

Connes–Kreimer Hopf algebras of flowcharts

• **The Connes–Kreimer renormalization procedure starts with the construction of a Hopf algebra generated by Feynman algebra.**

Here we will introduce its flowcharts version.

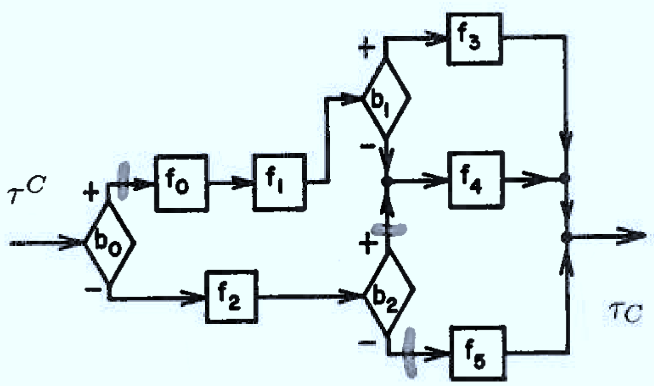
• **CUTS. Let τ be an oriented graph. Call** *a proper cut C* **of τ any partition of V_τ into a disjoint union of two non–empty subsets V_τ^C (upper vertices) and $V_{\tau,C}$ (lower vertices) satisfying the following conditions:**

(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.



τ^C
+
-
b_0
f_0
f_1
f_2
+
-
b_1
+
-
b_2
f_3
f_4
f_5
τ_C

Fix a set of labels $L = (L_F, L_V)$. Assume that $L_F = L_F^0 \times \{in, out\}$,

The isomorphism class of a decorated graph τ is $[\tau]$.

• DEFINITION. A set Fl ("flowcharts") of L–decorated graphs is called admissible, if the following conditions are satisfied:

(i) Each connected component of a graph in Fl belongs to Fl. Each disjoint union of a family of graphs from Fl belongs to Fl. Empty graph $\emptyset$ is in Fl.

(ii) For each $\tau \in Fl$ and each cut C of τ, τ^C and τ_C belong to Fl.

Fix a set of labels $L = (L_F, L_V)$**. Assume that** $L_F = L_F^0 \times \{in, out\}$**,**

The isomorphism class of a decorated graph τ **is** $[\tau]$**.**

• **DEFINITION. A set** Fl **("flowcharts") of** L**–decorated graphs is called admissible, if the following conditions are satisfied:**

(i) Each connected component of a graph in Fl **belongs to** Fl**. Each disjoint union of a family of graphs from** Fl **belongs to** Fl**. Empty graph** $\emptyset$ **is in** Fl**.**

(ii) For each $\tau \in Fl$ **and each cut** C **of** τ**,** τ^C **and** τ_C **belong to** Fl**.**

• **THE BIALGEBRA OF FLOWCHARTS:**

Fl **:= an admissible set of graphs,** k **:= a commutative ring.**

$H = H_{Fl}$**:= the** k**–linear span of isomorphism classes of graphs in** Fl**,**

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

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

sum over all cuts of τ.

• **CLAIM. (i)** m **defines on** H **the structure of a commutative** k**–algebra with unit** $[\emptyset]$**. Set** $\eta : k \to H,\ 1_k \mapsto [\emptyset]$.

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

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

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

• **CLAIM. (i)** m **defines on** H **the structure of a commutative** k–**algebra with unit** $[\emptyset]$**. Set** $\eta :\ k \to H,\ 1_k \mapsto [\emptyset]\,.$

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

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

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

• **THE HOPF ALGEBRA OF FLOWCHARTS. Existence and uniqueness of antipode will follow ([E–FMan]) if we introduce an appropriate grading 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:

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

3. REGULARIZATION AND RENORMALIZATION

• REGULARIZATION BY "MINIMAL SUBTRACTION". A typical example:

$\mathcal{A}$ **:= the ring of germs of meromorphic functions of** z **at** $z = 0$**;**

$\mathcal{A}_- := z^{-1}\mathbf{C}[z^{-1}]$,

$\mathcal{A}_+$ **consists of germs of regular functions at** $z = 0$**,**

$\varepsilon_{\mathcal{A}}(f) := f(0)$

For $f \in \mathcal{A}$**, the regularized value of** f **at** 0 **is** $\varepsilon_{\mathcal{A}}(f_+) = f_+(0)$ **where**

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

• CONNES–KREIMER RENORMALIZATION = a version of regularization that:

– Is performed simultaneously for an infinite family of functions indexeded by flowcharts;

– Uses "division by the collective pole part" in a noncommutative group in place of subtraction of an individual pole.

• **INPUT DATA:**

$\mathcal{H}$ **:= a Hopf K–algebra as above.**

$\mathcal{A}_+, \mathcal{A}_- \subset \mathcal{A}$ **a minimal subtraction unital algebra,** $\varepsilon_{\mathcal{A}} : \mathcal{A} \to K$.

$G(\mathcal{A})$ **:= 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

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

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

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

NB For any $x \in \ker \varepsilon$ **the latter sum contains only finitely many non–zero summands.**

• BIRKHOFF DECOMPOSITION: Collective pole and collective regular part.

If $\mathcal{A}$ is a minimal subtraction algebra, each $\varphi \in G(\mathcal{A})$ admits a unique decomposition of the form

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

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

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

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

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.

4. CUT–OFF REGULARIZATION AND ANYTIME ALGORITHMS

• Any regularization/renormalization scheme involves a deformation of initial problem: introduction of, say, a complex parameter z such that the infinite outcome of the initial problem can be treated as pole at $z = 0$, whereas outside $z = 0$ the deformed problem has a well defined solution.

• The simplest way of deforming an integral is to cut off the pole of integrand from the integration domain.

In theoretical computing, this might involve cutting computing time, storage capacity etc.

• Time cut–offs, accompanied by sober estimates of quality of outputs resulted in the notion of "Anytime Algorithms", cf. [Gr].

4. CUT–OFF REGULARIZATION AND ANYTIME ALGORITHMS

• Any regularization/renormalization scheme involves a deformation of initial problem: introduction of, say, a complex parameter z such that the infinite outcome of the initial problem can be treated as pole at $z = 0$, whereas outside $z = 0$ the deformed problem has a well defined solution.

• The simplest way of deforming an integral is to cut off the pole of integrand from the integration domain.

In theoretical computing, this might involve cutting computing time, storage capacity etc.

• Time cut–offs, accompanied by sober estimates of quality of outputs resulted in the notion of "Anytime Algorithms", cf. [Gr].

• THE CALUDE–STAY TIME CUT–OFF ARGUMENT:

(A) The runtime of the Kolmogorov optimal program at a point x of its definition domain is either $\leq cx^2$, or is not "algorithmically random" (Theorem 5 of [CalSt1]).

(B) Not "algorithmically random" integers have density zero for a class of computable probability distributions.

This last statement justifies the time cut–off prescription which is the main result of [CalSt1]:

• If the computation on the input x did not halt after cx^2 Turing steps, stop it, decide that the function is not determined at x, and proceed to $x+1$.

• One can generalize the statement (A) to arbitrary partial recursive functions in place of computation time.

Consider a pair of functions $\varphi, \psi : \mathbf{R}_{>0} \to \mathbf{R}_{>0}$ satisfying the following conditions:

a) $\varphi(x)$ and $\dfrac{x}{\varphi(x)}$ are strictly increasing starting with a certain x_0 and tend to infinity as $x \to \infty$.

b) $\psi(x)$ and $\dfrac{\psi(x)}{x\varphi(\psi(x))}$ are increasing and tend to infinity as $x \to \infty$.

The simplest examples: $\varphi(x) = \log(x+2)$, $\psi(x) = (x+1)^{1+\varepsilon}$, $\varepsilon > 0$.

In our context, φ will play the role of a "randomness scale". Call $x \in \mathbf{Z}^+$ *algorithmically φ–random,* if $C(x) > x/\varphi(x)$, where C is the (exponential) Kolmogorov complexity.

The second function ψ will then play the role of associated growth scale.

• PROPOSITION. Let f be a partial recursive function. Then for all sufficiently large x exactly one of the following alternatives holds:

(i) $x \in D(f)$, and $f(x) \leq \psi(x)$.

(ii) $x \notin D(f)$.

(iii) $x \in D(f)$, and $f(x)$ is not algorithmically φ–random.

5. REGULARIZATION AND HOPF RENORMALIZATION OF THE HALTING PROBLEM

• PLAN:

(a) Deforming the Halting Problem.

Recognizing, whether a number $k \in \mathbf{Z}^+$ belongs to the definition domain $D(f)$ of a partial recursive function f, is translated into the problem, whether an analytic function $\Phi(k, f; z)$ of a complex parameter z has a pole at $z = 1$.

(b) Choosing a minimal subtraction algebra.

Let $\mathcal{A}_+$ be the algebra of analytic functions in $|z| < 1$, continuous at $|z| = 1$. It is a unital algebra; we endow it with augmentation $\varepsilon_{\mathcal{A}} : \Phi(z) \mapsto \Phi(1)$. Put $\mathcal{A}_- := (1-z)^{-1}\mathbf{C}[(1-z)^{-1}]$, $\mathcal{A} := \mathcal{A}_+ \oplus \mathcal{A}_-$.

(c) Hopf algebra of a programming method.

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.

(d) 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** $p \in P(\mathbf{Z}^+, \mathbf{Z}^+)$**, is defined as** $\varphi_k([p]) := \Phi(k, f; z) \in \mathcal{A}$.

• A LESSON OF QUANTUM COMPUTING: Reduction of 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 an infinite 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 (but generally uncomputable) function** $g : X \coprod \{*_X\} \to X \coprod \{*_X\}$**, by** $g(y) := *_X$ if $y \notin D(f)$. **Define the map**

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

Clearly, 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** $D(\sigma_f) := (X \coprod \{*_X\}) \times D(f)$ **of the constructive world** $Y := (X \coprod \{*_X\})^2$ **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.

• THE KOLMOGOROV ORDER.

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** C_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**

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

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**

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

• THE HALTING PROBLEM RENORMALIZATION CHARACTER.

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

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

Then we have:

(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

[BaSt] **J. Baez, M. Stay.** *Physics, topology, logic and computation: a Rosetta stone.* **Preprint arxiv:0903.0340**

[CalSt1] **Ch. Calude, M. Stay.** *Most programs stop quickly or never halt.* **Adv. in Appl. Math., 40 (2008), 295–308**

[CalSt2] **Ch. Calude, M. Stay.** *Natural halting probabilities, partial randomness, and zeta functions.* **Information and Computation, 204 (2006), 1718–1739.**

[ConKr] **A. Connes, D. Kreimer.** *Renormalization in quantum field theory and the Riemann–Hilbert problem. I. The Hopf algebra structure of graphs and the main theorem.* **Comm. Math. Phys. 210, no. 1 (2000), 249–273.**

[E-FMan] **K. Ebrahimi–Fard and D. Manchon.** *The combinatorics of Bogolyubov's recursion in renormalization.* **math-ph/0710.3675**

[Gr] **J. Grass.** *Reasoning about Computational Resource Allocation. An introduction to anytime algorithms.* **Posted on the Crossroads website.**

[Ma1] **Yu. Manin.** *A Course in Mathematical Logic for Mathematicians.* **Springer Verlag, 2010. XVII+384 pp. (The second, expanded Edition, with collaboration by B. Zilber).**

[Ma2] **Yu. Manin.** *Classical computing, quantum computing, and Shor's factoring algorithm.* **Séminaire Bourbaki, no. 862**

(June 1999), Astérisque, vol 266, 2000, 375–404. quant-ph/9903008.

[Ma3] **Yu. Manin.** *Renormalization and computation I. Motivation and background.* **math.QA/0904.492**

[Ma4] **Yu. Manin.** *Renormalization and computation II: Time cut–off and the Halting Problem.* **math.QA/0908.3430**

[Sc] **D. Scott.** *The lattice of flow diagrams.* **In: Symposium on Semantics of Algorithmic Languages, Springer LN of Mathematics, 188 (1971), 311–372.**