

---

# Abstract Chaitin's theorem and its methodological consequences

NIKOLAY N. NEPEIVODA

---

**ABSTRACT.** Abstract forms of Kolmogoroff's complexity, Chaitin and Gödel's theorems are stated. They are used to analyze numerous methodological issues: Kant's Third antinomy, Parkinson's law of committee, cooperative creative activity, multi-language programming, benevolence to other's views, dilemma of deism–atheism.

*Keywords:* Kolmogoroff complexity, Chaitin theorem, Gödel theorem, Kant antinomy, Parkinson law

Chaitin's theorem of unknowledgeable and Gödel's theorem of incompleteness are of great importance for philosophy of science. Numerous works around them are based on supposition that functions considered are algorithmic and objects are constructive (more specifically natural numbers). First of all we make some kind of reverse analysis to extend them into abstract domains and for wide class of theories. In essence we show that there are very weak assumptions on derivability of true formulas but somewhat stronger for underivability of false formulas.

## 1 Abstract computations

DEFINITION 1. Let there is a finite set of letters  $\mathbf{A}$ . Lists of signature  $\mathbf{A}$  are defined inductively:

1. Empty list  $()$  is denoted  $NIL$ .
2. Each letter is an atom.
3. If  $a_1, \dots, a_n$  are lists or atom then  $(a_1 \dots a_n)$  is a list.

A natural number  $n$  is represented by the list  $(NIL \dots NIL)$  ( $n$  members). Thus 0 is  $NIL$ , 1 is  $(NIL)$ , 2, is  $(NIL NIL)$  and so on.

**DEFINITION 2.** Functional signature is a finite set of atoms, including `cons`, `car`, `cdr`, `lh`, `members`, `id`, `concat`, `quote`, `arity`, `expand`, `join`, `perm`, `comp`, `const`.

Interpreter functional signature contains in addition to the above `turing`, `ifnil`, `ifatom`, `iflist`, `iffunction`, `equal`. Turing one contains also `eval`, full Turing one adds `search`. All these atoms are *elementary functions*. Lists in a functional signature are called *expressions*.

We interpret as a function the last member of a list.

**Comment.** It is for we consider Turing-incomplete languages where we cannot define a function to add an element to the tail of a list.

Some lists are functional ones. Some non-functional lists are convertible and can be computed.

**DEFINITION 3.** Let  $F$  is a functional list,  $E$  is any list. We can add integer indices which are written in the same string.

Elementary functions are functionals. Lists  $(E F \text{ expand})$ ,  $(E1 E2 F \text{ join})$ ,  $(F1 E F2 \text{ comp})$ ,  $(E \text{ const})$ ,  $(E1 E2 \text{ turing})$ ,  $(F1 F2 E \text{ ifnil})$ ,  $(F1 F2 E \text{ ifatom})$ ,  $(F1 F2 E \text{ iflist})$ ,  $(F1 F2 E \text{ iffunction})$ ,  $(E E1 \text{ eval})$ ,  $(F F1 \text{ search})$  are functionals.

Now a computational semantic of functions is defined. `arity` is applied to a functional and gives a number of arguments of a resulting function.

$(\text{arity } \text{arity}) = 1.$

$(E1 E2 \text{ cons})$  computes both arguments and makes a list with the head  $E1$  and the remaining part  $E2$ .

$(E \text{ car})$  computes  $E$  and extracts its first element.

$(E \text{ cdr})$  computes  $E$  and removes its first member.

$(E \text{ lh})$  gives the number of symbols in the value of  $E$ . Each atom and each bracket is a single symbol.

$(E \text{ members})$  gives the number of elements in the value of  $E$ .

$(E \text{ id})$  gives a value of  $E$ .

$(E1\ E2\ \text{concat})$  joins two lists together.  
 $(E\ \text{quote})$  don't computes  $E$  and updates it as is.  
 $(F\ \text{expand})$  adds a fictive argument to the tail of arguments of  $F$ .  
 $(E1\ E2\ F\ \text{join})$  computes  $E1$ ,  $E2$  and if their values are numbers in  $1 \leq E_i \leq (F\ \text{arity})$  diminishes number of arguments of  $F$  by 1 glueing arguments with those numbers.  $E1$  is remaining in the list of arguments.  
 $(E1\ E2\ F\ \text{perm})$  is analogous permutating two arguments.  
 $(E\ F1\ F2\ \text{comp})$  substitutes  $F1$  for argument of  $F2$  with number  $E$ .  $1 \leq E \leq (F2\ \text{arity})$ .  $((E\ F1\ F2\ \text{comp})\ \text{arity}) = (F1\ \text{arity}) + (F2\ \text{arity}) - 1$ .  
 $(E\ \text{const})$  is a function of arity 1 always giving value of  $E$ .  
 $(E1\ E2\ E3\ \text{turing})$  computes  $E2$ , which is to be a functional, and then performs  $E3$  steps of its application to  $E1$  ( $E3$  is to have a number value) and gives a list  $(E4\ E5)$ , where  $E4 = 0$ , if computation had been finished on or before step  $E3$ , and 1 else.  $E5$  gives a result of (partial) computation. We accept that  $(E1\ 0\ E3\ \text{turing}) = (1\ E1)$ .  
 $(E\ F1\ F2\ \text{if}[\text{property}])$  computes  $E$  and if its result has a desired property gives  $F1$ , else  $F2$ . Arity of two functionals are to be equal.  
 $(E1\ E2\ \text{equal})$  gives 0, if results are literally the same, 1 else.  
 $(E1\ E2\ \text{eval})$  computes its arguments, second one is to be a functional, then applies this function to the value of the first argument.  
 $(F\ F1\ \text{search})$  finds such tuple of values of arguments for  $F$ , for which  $F$  is equal to 0, and applies  $F1$  to a found values. Arities of functions are to be equal.

A list is *convertible*, if there is a subexpression which is not functional of the form  $(E1\ \dots\ En\ F)$ , where  $(F\ \text{arity}) = n$ .

There can be any number of extra elementary functions in our system. The only condition is that each function have a well defined computational semantics (not necessary algorithmic). Thus we defined a kernel language for different kinds of algorithmic and non-algorithmic computations (e. g. hyperarithmetic or computations on an algebraic structure).

PROPOSITION 1. (*limited  $\lambda$ -abstraction*) *Let us enrich our language by variables  $x1, \dots, xn$ . Then for any list  $E[x1, \dots, xn]$  can be constructed a functional  $FT$  s.t.  $(E1\ \dots\ En\ FT) = E[E1, \dots, En]$ .*

PROOF is purely technical.  $\square$

PROPOSITION 2. *There holds a fixed point theorem in each Turing system: for each functional  $F$  there is such  $E$ , that for all  $E1$*

$$(E1\ E\ F) = (E1\ E\ eval).$$

PROOF.

$$\begin{aligned} LUR &= (1\ (eval\ const)\ (2\ (NIL\ const)\ cons\ comp)\ comp) \\ XXU &= (1\ 2\ (1\ (1\ LUR\ cons\ comp)\ cons\ comp)\ join); \\ LXF &= (1\ (XXU\ const)\ F\ comp) \\ YF &= (1\ LXF\ XXU\ comp). \end{aligned}$$

YF is a fixed point.  $\square$

PROPOSITION 3. *(Turing completeness) Turing systems allow to express any partial recursive function.*

PROOF. By fixed point and conditionals we can construct McCarthy recursive schemes.

Note now that `eval` is definable through `turing` and `search`. `eval` is called a universal function, `turing` is an interpreter, `search` is a search operator. No other dependencies hold for these three operators. Primitive recursive functions have an interpreter without search and universal function. Recursive schemata on real numbers and their lists with a signature  $\{0.0, 1.0, =, >, +, *\}$  have universal function and interpreter but no search. Hyperarithmetical functions on real numbers have no search and no interpreter, only a universal function. Adding search we get no interpreter. Adding search to initial elementary functions gives no interpreter and no universal function.  $\square$

## 2 Generalization of Kolmogoroff complexity

DEFINITION 4. *Complexity* of an object relatively to a computational system is a minimal length of an expression which evaluates

to our object. If a system is turing one, complexity is called kolmogoroff one. Complexity of an object  $x$  in a system  $\Sigma$  is denoted  $(x \mathbb{K}_\Sigma)$ .

If system is defined by a context it is omitted.

DEFINITION 5. Let there are two computational systems  $\Sigma_1$  and  $\Sigma_2$ . Coding  $\text{CODE}[a]$  of language of one system inside language of other is *regular*, if

$$(\text{CODE}[a] \text{ lh}) \leq k \cdot (a \text{ lh}) + C_1$$

$C_1$  — is a constant,  $k$  is a *coding factor*.

$\Sigma_1$  is interpreted in  $\Sigma_2$ , if there is a regular coding and a function  $\text{Int}$  such that

$$\exists n (\text{CODE}[E] \text{ Int turing}_2) = (0 \text{ CODE}[a]) \iff E = a.$$

$\Sigma_1$  is translated into  $\Sigma_2$ , if there is a regular coding and a function  $\text{Trans}$  such that

$$(\text{CODE}[E] \text{ Trans eval}_2) = \text{CODE}[a] \iff E = a.$$

THEOREM 1 (KOLMOGOROFF'S THEOREM). *If  $\Sigma_1$  is interpreted or translated into  $\Sigma_2$  and  $k$  is a coding factor, then  $k \cdot (a \mathbb{K}_1) \leq (\text{CODE}[a] \mathbb{K}_2) + C$ .*

PROOF is obvious. □

This theorem generalizes up to wide class of systems and codings (including Turing-incomplete and non-algorithmic) a theorem of Kolmogoroff on invariancy of complexity up to additive constant.

### 3 Generalization of Chaitin theorem

Let there is a theory  $\text{Th}$ , with definable predicates 'To be a natural number', = and < for natural numbers, constants 0, 1 and functions +, \*,  $\uparrow$ , the last one is a power function. Elementary arithmetical formulas are relations of two expressions in this vocabulary. Then we say that this theory contains natural numbers.

Let there is a full turing system  $\Sigma$  with functionals to test whether this list is a proof of a given formula in some regular coding, to

extract a proved theorem from a proof code and to substitute an object of  $\Sigma$  (not necessarily a number) for a free variable of a formula and to compare two formulas textually.

DEFINITION 6. A theory is chaitin-correct w.r.t.  $\Sigma$ , if are expressible: a notion  $(E \ E1 \ eval)=a$ , a function  $(a \ lh)$ , all true formulas  $(a \ lh) = n$  are provable, all closed true elementary arithmetical formulas are provable, and any closed false formula of the form  $\neg(E \ E1 \ eval)=a$  is not provable.

Each chaitin-correct theory is consistent. A simplest such theory  $\mathbf{Ar}_0$  is given by the following axioms:

$$\begin{array}{ll} \forall x (x + 0 = x) & \forall x, y (x + (y + 1) = (x + y) + 1 \\ \forall x (x * 0 = 0) & \forall x, y (x * (y + 1) = (x * y) + x \\ \forall x (x \uparrow 0 = 1) & \forall x, y (x \uparrow (y + 1) = (x \uparrow y) * x. \end{array}$$

THEOREM 2. *There is a number  $C$  in any chaitin-correct theory such that  $(a \ \mathbb{K}) > C$  is not provable for any  $a$ .*

PROOF. A formula expressing  $(E \ E1 \ eval)=a$  is denoted  $R(p, x, a)$ . Then a statement  $(a \ \mathbb{K}) > C$  can be formulated as follows:

$$\forall x \forall p ((x \ p) \ lh) < C + 1 \supset \neg R(p, x, a).$$

If  $(a \ \mathbb{K}) < C + 1$  holds, then this formula is not provable inside  $\mathbf{Th}$ , because elsewhere it would be provable a false statement  $((x0 \ p0) \ lh) < C + 1 \ \& \ \neg R(p0, x0, a)$  and thus a false formula  $\neg R(p0, x0, a)$  for some  $((x0 \ p0) \ lh) < C + 1$ . Let show this and by the way construct a Chaitin's constant.

Let a functional  $K$  finds for each  $C$  searches a proof of a formula  $(a \ \mathbb{K}) > C$  by brute force and if such proof is found gives  $a$ . Let a length of code for this functional is  $k$ . Let the quantity of different atoms in our system is  $m$ . Then there is such  $C_0$ , that  $m^{C_0} > k * C_0$ . This  $C_0$  can be taken as a Chaitin's constant. Let  $(a \ \mathbb{K}) > C_0$  were provable. Then  $K$  would find such  $a0$ . But really  $(a0 \ \mathbb{K}) \leq C_0$  and thus  $((x0 \ p0) \ lh) < C_0 + 1 \ \& \ \neg R(p0, x0, a0)$  is not provable for some  $p0, x0$ . But  $\forall x \forall p ((x \ p) \ lh) < (C_0 + 1) \supset \neg R(p, x, a0)$  implies  $((x0 \ p0) \ lh) < (C_0 + 1) \supset \neg R(p0, x0, a0)$ .  $((x0 \ p0) \ lh) < (C_0 + 1)$  is provable by correctness, therefore is provable  $\neg R(p0, x0, a0)$ . Contradiction.  $\square$

This form of Chaitin's theorem does not demand computability of a system complexity is defined w.r.t. It uses search function essentially. It can be applied also for systems with infinite basic data type but with finite base of explicitly given atoms. Then complexity of some objects can be infinite (e.g.  $\pi$  in a system for algebraic operations on real numbers).

#### 4 A generalized Gödel incompleteness theorem

Now we consider and generalize the Gödel incompleteness theorem in the form of Rosser [2]. Let us give some auxiliary definitions.

DEFINITION 7. Restricted quantifiers are formulas of the form

$$\forall x ((x \text{ lh}) < \mathbf{n} \supset A(x)), \quad \exists x ((x \text{ lh}) < \mathbf{n} \& A(x)).$$

A formula  $P(x)$  is *limitedly correct* in the theory  $\mathbf{Th}$ , if from provability of  $\exists x ((x \text{ lh}) < \mathbf{n} \& P(x)) \vee B$  follows provability of  $P(\mathbf{a})$  for some  $(\mathbf{a} \text{ lh}) < \mathbf{n}$  or provability of  $B$  itself.

DEFINITION 8. A theory is *Gödel-correct* if a predicate  $<$  is expressible for natural numbers; all closed true formulas of the form  $(\mathbf{a} \text{ lh}) < \mathbf{n}$  are provable; there is some coding for formulas; there is a formula expressing 'p is a proof of  $A(a)$ '  $Proof(p, \text{CODE}[A], a)$ ; there is a functional to compute code of negation of a formula by its code  $\text{Neg}$ ; if  $A(a)$  is provable, then  $Proof(p, \text{CODE}[A], a)$  is provable for some  $p$ ; a weak Gödel rule

$$(1) \quad \frac{Proof(p, \text{CODE}[A], a)}{A(a)}$$

is admissible and  $Proof(p, \text{CODE}[A], a)$  is limitedly correct for all  $A, a$ .

THEOREM 3. *If a theory is Gödel-correct it is incomplete.*

PROOF. Consider a formula

$$(2) \quad \begin{aligned} & \forall x ((Proof(x, z, z) \supset \\ & \exists y ((y \text{ lh}) < (x \text{ lh}) \& Proof(y, (z \text{ Neg}), z))) \& \\ & \exists x ((Proof(x, (z \text{ Neg}), z) \\ & \& \neg \exists y ((y \text{ lh}) < (x \text{ lh}) \& Proof(y, z, z)))) \end{aligned}$$

Substitute in it its code  $R$ . Then if the formula

$$(3) \quad \begin{aligned} & \forall x ((Proof(x, R, R) \supset \\ & \exists y ((y \text{ 1h}) < (x \text{ 1h}) \& Proof(y, (R \text{ Neg}), R))) \& \\ & \exists x ((Proof(x, (R \text{ Neg}), R) \& \\ & \neg \exists y ((y \text{ 1h}) < (x \text{ 1h}) \& Proof(y, R, R)))) \end{aligned}$$

is provable, we take  $a_0$  with provable  $Proof(a_0, R, R)$ . Due to limitedly correctness of  $Proof$  and by the first conjunctive subformula there is such  $(a_1 \text{ 1h}) < (a_0 \text{ 1h})$ , that  $Proof(a_1, (R \text{ Neg}), R)$  is provable. Then by a rule (1) is provable a negation of (3) and our theory is inconsistent and proves everything. So it is not Gödel-correct.

If a negation of (3)

$$(4) \quad \begin{aligned} & \exists x ((Proof(x, R, R) \& \\ & \neg \exists y ((y \text{ 1h}) < (x \text{ 1h}) \& Proof(y, (R \text{ Neg}), R))) \vee \\ & \forall x ((Proof(x, (R \text{ Neg}), R) \supset \\ & \exists y ((y \text{ 1h}) < (x \text{ 1h}) \& Proof(y, R, R)))) \end{aligned}$$

is provable then there is such  $b_0$  for which  $Proof(b_0, (R \text{ Neg}), R)$  is provable. From first disjunctive part follows

$$\exists x ((x \text{ 1h}) < b_0 + 1 \& (Proof(x, R, R))).$$

Applying limitedly correctness we get provability whether (3), which is contradictory, or the second disjunctive part. Then we get a contradiction analogously to the first part of proof.  $\square$

## 5 Philosophical consequences

Kant's Third Antinomy (of Freedom) can be substantiated precisely if complexity of a human is lower than complexity of the Universe. Parkinson's law of committee (decision of committee is more moronic than decision proposed of its stupidest member) can be proved precisely. One of paradoxes arising while applying precise Computer Science to real Informatics can be solved. It is known that Kolmogoroff's complexity is invariant up to ADDITIVE constant  $L$ . Using Chaitin's limit we can prove that the fixed constant  $L$  can substantially decrease the actual possibilities of programmer. Interrelation

of Chaitin and Orevkov theorems yields that high level person can make things which cannot be understood by plain thinkers but to implement his/her insights plain thinking is often necessary. Some peculiarities of Chaitin's limit if person's mind is not Turing complete are considered.

## 6 Algorithmic randomness and Kant's Third Antinomy

So any formalism has limits such that upper them it cannot state a complexity of an object and thus cannot correctly comprehend and understand it. So an argumentation with complexity upper than Chaitin's limit for a person is understood by completely chaotic and illogical. But this is not the worst case. If such person tries to comprehend the arguments by cutting out all which cannot be placed in his/her head he/she gets an illusion of understanding together with completely wrong image of percept.

Chaitin [3] noted out that now existence of unknowledgeable is well substantiated and even proved. Each position based on supposition that human mind is omnipotent in principle is not even an opinion now. Our generalization of Chaitin theorem shows how weak premisses are sufficient for Chaitin's limit is existent. We do not need here to claim that human is a finite system which had been used in earlier demonstrations. This together with an observed harmony of the world substantiated theism in very high degree [4]. At the same time this shows that it is impossible to prove or to refute existence of God.

For finer methodological consequences it is reasonable to accept finiteness of a human (as for example in [7]). Thus because complexity of the Universe is much higher than one of a human and of the humanity (even in supposition that joining humans join only knowledge but not their ignorance). But incognizable can sometimes be partially appreciated. It is known that objects with big Kolmogoroff complexity are comprehended as random.

Kolmogoroff studied algorithmic randomness for infinite sequences (complexity of initial segment of a sequence will be same as its length up to additive constant). We are to define randomness of a finite object from the point of view of Chaitin's limit and his

considerations in [4, 5]. This is randomness relative to a concrete object or subject processing information.

*An object is random for a processor if its complexity is larger than processor's Chaitin's limit.*

Now we'll prove a proposition equivalent to Kant Third Antinomy [8] and even more strong, expressing it in the language of current science.

*Human cannot state whether our Universe is deterministic or there is a necessary randomness in it.*

Let the Universe be deterministic. Then a complexity of the algorithm initialized during world's creation is higher than Chaitin's limit of humanity. Thus humanity cannot comprehend a Word's idea as a whole and complete entity. Deterministic world is understood as random one.

**Note!!!** We are not creationists here. World creation would be a natural process for example as a garbage of a super-civilization during re-creation or transformation of its own World (S. Lem: From Einsteinian to Testan Universe. In Polish).

Let our World be indeterministic. If we were proved this we were proved that complexity of our World is higher than Chaitin's limit of our civilization. This is a contradiction.

Thus problem whether our Universe is deterministic is a pseudo-problem from the point of view of pure exact knowledge. We are free to choose a theory which in the moment is a best fit for 'practice' and is a better representation of objects in view.

Therefore it is unacceptable to advertise results of our science as 'scientific truth'. They are to be re-verified by an alternative theory. This is a strong opposition for postmodernistic 'tyranny of truth'. We cannot lay our responsibility on arms of Science or God.

## 7 Parkinson's law

Let there is a committee which is to work out a decision understandable for all its members for each could meaningfully vote 'yea' or 'nay'. In this case Chaitin's limits of committee members are to

be reduced to minimal one because else some of members cannot understand a proposal. So a weak Parkinson's principle is substantiated:

*Weak Parkinson's law:*

*Decision of a committee is no more adequate that one which could make the least competent of its members himself.*

But the reality is more crude. Each committee member has different competentions in different domains. So we need to introduce a matrix of limits. If two limits of persons are  $C_i$  and  $C_j$ , complexities of translations from one system of notions into an other are  $K_{ij}$  and  $K_{ji}$ , then maximal complexity of a decision of each of them understandable by both is  $C_{ij} = \min \{C_i - K_{ji}, C_j - K_{ij}\}$ : a limit of i-th person for understanding of j-th. Thus even not taking into account non-uniformity of knowledge inside a Chaitin limit we get the following upper bound:  $\min_{i,j} C_{ij}$ . We substantiated the following

*Strong Parkinson's law:*

*Decision of a committee is more moronic that a decision which could make the most moronic of its members himself.*

In Venice and Rome important decisions were delegated to a truthful person which had been made fully responsible for its realization and consequences...

## 8 Chaitin limit and paradox of inventor (Orevkov theorem)

There is at least one more quality of mind orthogonal to brute force which can lead to relatively large Chaitin's limit. This is ability to master complex notions.

**Orevkov theorem (1968):** Indirect proof in logic can be in the tower of exponents times shorter than direct one.

Orevkov's theorem is a precise partial case of a general paradox of inventor formulated by Gy. Polya:

To prove a simple statement we are often to use complex intermediate notions. To prove a weaker and 'simpler' statement can be much more harder than to prove more strong and complex one.

Gy. Polya pointed out and partially explained this paradox w.r.t. inductive proofs. Orevkov substantiated that it is a fundamental property of thinking.

Using high order notions we can jump far away behind Chaitin's limit of crawling persons. This substantiates a genial insight of D. Hilbert that ideal notions are necessary to obtain non-trivial practical (real) results.

American scientist M. Furman wrote (private communication discussing my preliminary notes on Chaitin's limit): 'Non-equivalence (not considering purely theoretical notion of Kolmogorov complexity, but from the point of view of real application) is defined by resources: size of memory and execution time.'

Theoretically we have two binary properties: is memory finite or is time finite. But seeing one step deeper we understand that there is a uniform restriction for some class of examples'.

These arguments do not disturb our basic considerations and only show that real situation is even more fine and interesting. It is known that primary resource of human defines his/her logic (linear logic is logic of money < intuitionistic one is logic of knowledge, nilpotent one is logic of time and so on). Of course it can restrict Chaitin's horizon even more substantially than Kolmogoroff complexity.

M. Furman also proposed an example showing interconnections of Chaitin's limit with inventor's paradox. If a person mastered a high-level method he can say something like to Furman's objection: 'It is very easy to construct a translator having the precise definition of a language'. But method of formal semantics itself cannot be treated as a simple one. And it is known how hard is to write out a formal definition of a semantic.

Evgeny Kochurov pointed out (private communication) that usually those who cannot comprehend complex notions but have a big operative memory can build long and relatively complex first-order compositions. Those who excellently appreciate methods can find excellent critical points but poorly analyses a crawling process how to go from one critical point to next one. So those two are complementary and can excellently assist one another if each person is used

according to his/her strong sides. So we transferred to a problem how to avoid Parkinson's law.

## 9 Consequences for organisation of creative work: How to avoid Parkinson's law?

There is an interesting example which seems to be a strong counterexample to Parkinson's law. Each bee, termite or ant acts like finite automaton with a fixed program and low memory. Nevertheless a general behaviour of nest become very complex and adaptive. Moreover ants for example demonstrate more complex forms of integration and system behaviour. Remember ant empires joining in the single net thousands of nests which have intensive exchange of information, people and genetic material (trade points and exchange of nymphs).

We apply here an analogy from logic. Von Neumann's theory of self-reproducing automata shows how to compose an upcoming system from uniform units with extremely simple behaviour. Thus a good organization of morons which cannot understand even loops can generate recursions and high level constructions.

How is it possible? It is because cooperation itself is performed by strict simple automata rules. This analogy is used in neuron nets in such domains as pattern recognition in cases when there are no precise algorithms. Well trained neuron net mistakes sometimes but rarely. And nobody knows why.

Ideology of crowdsourcing tries to transfer this experience into human society. But as for neuron nets here we get no creativity<sup>1</sup>. How to introduce it?

As usually direct and obvious decision — to make automata stochastic or indeterministic — fails here. Such approach to creation process is fantastically ineffective.

So we come to a tough consequence for human collectivities. Committee consisting from equal and free creative persons is im-

---

<sup>1</sup>In Russian there are two words for English 'creativity'. Креативность (creativity) means invention of something new only to be new without real values and goals. Творчество means creation of new and useful things. This is why 'creative class' is appreciated by Russians as class of uppity, spiritually and really impotent egocentric persons.

potent. Potent can be at least two-level structure. Interactions are strictly formalized on first level and for connections between first and second level. In contrary interactions on second level are bounded by clear and ruthless rituals but never formalized. They are diminished to a reasonable minimum. Upper level is responsible for creative decisions and lower for their realization. It is often possible to implement an idea inside a rigid structure but never is possible to get a new idea here.

We have here another 'counterexample': freesofters. This seems to be a conglomerate of free creative individuals which interact very informally. But this is not the case. They curse and laud one another very informally but their interactions in coding, bug processing, documentation and so on follow strict rules. So I cannot say that they are 'free persons' in vulgar sense of this word. They are free individuals having real goals and values and voluntarily sacrificing some 'freedoms' for those high valuables. They can be an embryo of a structure which can save humanity and some real achievements of current ill civilization after its inevitable death.

And now in a cold water. A community of freesofters can be so effective because almost all of them are involved into really non-creative problems of coding according to existing algorithms and architects, debugging and developing earlier projects. But this community has also an ecological niche for really creative persons.

**Warning.** A society based on freesofters-like libertarian principles will ruthlessly apply 'measures of humanitarian defence' (see e.g. A. A. Rosoff 'Confederation Meganesia') and suppress minorities which wish to claim their rights in manner restricting other people's rights and common values. It may be necessary to survive against mindless hordes of 'free vultures'.

Furthermore collective intellect of best algebraists allowed to solve a problem of classification of finite groups [9]. But interaction of professional pure mathematicians is so deeply ritualized<sup>2</sup> that this example is a verifying example for us.

These examples allow us to make principle of committee more precise. Committee must elaborate a decision. Such decision will inevitably be a compromise e.g. a mixture of unpleasant and useless.

---

<sup>2</sup>And not formalized, in contrary to common prejudice.

Creative persons try to find a solution. They do not try to cut it according to lower level of their understanding. In contrary, people develop an other's people nice idea even they do not appreciate it as a whole and often find new aspects of it. So a good organized creative storming can lead to a valuable results. High level people know how useful is a discussion of equal in spirit and mind persons (but not those nominated by an institution).

Collective creative work is development and transformation of new ideas without 'full comprehension'.

How to increase effectivity of this storming?

1. Sacrifice sacred cows.
2. Make hidden conceptual contradictions visible.
3. Don't pronounce 'universal and indisputable truths' (BLA-GOGLUPOSTI (in Russian) I don't know an analogy in English).

All these three points contradict to politcorrectness and other liberal taboos.

## 10 Chaitin limit and programming languages

Formally complexity of programs in different PL is equivalent up to additive constant (Kolmogoroff theorem). Practice shows the opposite: program written by adequate tools can be 50 times shorter than in 'universal' Java or C#. Why?

Kolmogoroff's theorem (1) states that  $k \cdot (a \mathbb{K}_1) \leq (\text{CODE}[a] \mathbb{K}_2) + C$  where  $k$  is equal to 1 if we consider standard programming codes. Constant  $C$  is a length of a translator program for the second language written in the first language. To write it eats almost all Chaitin's limit of a programmer.

Therefore we have an excellent and precise demagogic answer on a moronic and demagogic question very often posed to ones who did something by 'exotic' language: 'Is it possible to write the same in C# or Java?':

— Of course. It is possible to write all in the language of Turing machines, if you prefer.

Thus theoretical equivalence sometimes means practical incomparability.

This analogy works in other domains also. If we do not master a language of a concrete domain we can *in principle* to understand constructions and arguments but it is necessary to build in our mind a 'translator' into our paradigm. Its complexity can be so high that it leaves almost no resources to analyze the argumentation.

Another warning. If you know many languages but have no background fundamental knowledge in your head you work worse than blind coder. Multi-tool method is effective only when a person masters a meta-knowledge, meta-method and a basis of notions.

So fundamental knowledge is that which forms a system in a brain. Foundation of a system must be stable. It consists of a basis of relatively simple notions (keystones) amalgamated by a lot of relation and properties which show their interrelations gains, shortcomings and restrictions. It is ideal if in result a person sees restrictions of his/her system as a whole.

And there is one more bad side. Many people simply cannot appreciate complex (algorithmic) constructions such as recursions and even loops. They have no universal algorithm in their head. Here Chaitin's limit is 0 and this person simply can see nothing.

### Final remark

*It is false that clever one works faster than more stupid one. A stupid person never can understand what does a clever one and never can make the same work.*

## 11 Benevolence to other's views

A problem of co-existence of different views is madly contaminated by 'tolerance' originated in the fundamental mistake of J. S. Mill: he declared freedom of opinions instead of freedom of argumentation. He simply could not imagine that every irresponsible and moronic cry will demand rights and honors because it is an 'opinion of a free person'.

This goes deeper to BLAGOGLUPOST of Voltaire's 'I hate your opinions, but I would die to defend your right to express them'. We see that there are too much people who accept no counter-arguments

against their opinions but are ready to kill each who criticizes them. We see that there are too much people and institutions which substantiate their opinions not by argumentation but by direct lie and manipulations (e.g. neo-liberals, neo-cons, fundamentalists, juvenile justice ...).

*Principle of benevolence to other's views.*

Remember that The Truth is inaccessible to you and to any other human. Thus say confronting other views.

I do not agree with your views but you argue in their favor honestly and earnestly. Thus I will defend your right to proclaim them, to substantiate them and to distribute them. In the same time I declare full and unrestricted right of me and any other person to criticize them, to find weak points in your argumentations and maybe lie and manipulations.

This obligation is ended when your sights become refuted or you are caught on lie or manipulations (sophistic or psychologic).

In the first case you remain an honest person for me and I will defend you against any attempts to punish you for error itself (but not for its consequences). If you will be so brave to recognize you have been mistaken I will help you to correct it and its consequences and you will become greater in my eyes.

If you will be caught on dishonored tricks all my responsibility will end. I will support the toughest of possible legal punishments for you because spiritual poison is more mortal than material.

## **12 Methodological argument for deism**

Chaitin's theorem showed that Kant was right stating that our intellect cannot solve a problem of God's existence. So we have the following consequences.

1. Existence of God is a pseudoproblem from scientific point of view and you must take your own decision here.
2. It is unacceptable to cry that science rejects God (and equally that science proves God's existence).

3. It is inadmissible to make any scientific consequences from existence or non-existence of God.
4. It is acceptable to analyze this problem methodologically.

So the problem of deism or atheism is a methodological problem. Stating a rational definition of God as *The Truth*, as *the unified highest law of both nature and spirit which is beyond all worlds and all times* we are inspired to find unity in difference, high level unifying notions and principles for realizations which seem to be not connected for plain thinking, or even contradictory though both existing. It inspires us to develop ourselves both intellectually and spiritually and to keep these different sides and our material being in harmony.

In contrary atheism demotivates us to idolize and adore our imperfect plain reasoning and our restricted knowledge.

## References

- [1] CHAITIN, G. J., Information-theoretic limitations of formal systems, *J. of the ACM* 21:403–424.
- [2] KLEENE, S. C., *Introduction to metamathematics*, Princeton, NJ: Van Nostrand, 1964.
- [3] CHAITIN, G., *Meta Math!: The Quest for Omega*, Pantheon Books, 2005.
- [4] CHAITIN, G., *Mathematics, Complexity and Philosophy, Editorial Minda*, 2011.
- [5] CHAITIN, G., Randomness in arithmetics, *Scientific American* 9:56–68, 1988.
- [6] KRITCHMAN, S. and R. RAZ, The Surprise Examination Paradox and the Second Incompleteness Theorem, *Notices of the American Mathematical Society*, 57(11):1454–1458, 2010.
- [7] RAATIKAINEN, P., On interpreting Chaitin's incompleteness theorem, *Journal of Philosophical Logic* 27:569–586, 1998.
- [8] KANT, I., *The Critique of Pure Reason*. <http://www.gutenberg.org/ebooks/4280>
- [9] ASCHBACHER, M., The status of the classification of the finite simple groups, *Notices of the American Mathematical Society* 51:736–740, 2004.