PRACE IPI PAN ICS PAS REPORTS

Stanisław Ambroszkiewicz

CONTINUUM AS A PRIMITIVE TYPE

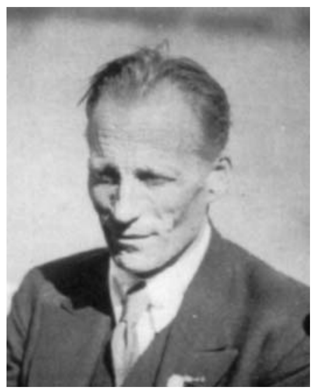
Nr 1034

INSTYTUT PODSTAW INFORMATYKI POLSKIEJ AKADEMII NAUK
INSTITUTE OF COMPUTER SCIENCE POLISH ACADEMY OF SCIENCES
ul. Jana Kazimierza 5, 01-248 Warszawa http://www.ipipan.waw.pl

Stanisław Ambroszkiewicz

Continuum as a primitive type

Nr 1034



Luitzen Egbertus Jan Brouwer
(1881 – 1966)

Some of the results presented in the paper were inspired by the work of L. E. J. Brouwer

Institute of Computer Science
Polish Academy of Sciences

Warsaw, October 2015

Pracę zgłosił Prof. dr hab. inż. Wojciech Penczek

Adres autora: Instytut Podstaw Informatyki PAN
ul. Jana Kazimierza 5
01-248 Warszawa
Polska
E-mail: sambrosz@ipipan.waw.pl

Symbol klasyfikacji rzeczowej MSC 2000: 03D99, 68Q05

Na prawach rękopisu
Printed as manuscript

Nakład 100 egzemplarzy. Papier kserograficzny klasy III. Oddano do druku w październiku 2015 r. Wydawnictwo IPI PAN. ISSN 0138-0648

Abstract

Continuum as a primitive type

The paper is the revision, extended and full version of the short (6 pages) preliminary presentation of the grounding of the notion of Continuum given in Section 6 of the paper *Types and operations* ICS PAS Report No. 1030 (also at `http://arxiv.org/abs/1501.03043`). Here, primitive types (corresponding to the intuitive concept of Continuum) are introduced along with primitive operations, constructors, and relations.

The paper is also at arXiv `http://arxiv.org/abs/1510.02787`

Keywords: Continuum, types, semantics, foundations of Mathematics

Streszczenie

Continuum jako typ pierwotny

Praca jest znacznym rozszerzeniem i uzupełnieniem Rozdziału 6 pracy *Types and operations* Prace IPI PAN Nr 1030 (również na `http://arxiv.org/abs/1501.03043`). Tutaj nowe pierwotne typy (odnoszące się do intuicyjnego pojęcia Continuum) są wprowadzone razem z pierwotnymi operacjami, konstruktorami oraz relacjami.

Słowa kluczowe: Continuum, teoria typów, podstawy Matematyki

1 Introduction

In the XIX century and at the beginning of the XX century there was a common view among the mathematicians that Continuum is different than natural numbers and cannot be reduced to them, that is, Continuum cannot be identified with the set of the real numbers, or in general with a compact connected metric space. Real numbers are defined on the basis of rational numbers (for example, as equivalence classes of Cauchy sequences), and rational numbers on the basis of natural numbers.

The following citations support this view.

- D. Hilbert [11]: *the geometric continuum is a concept in its own right and independent of number.*
- E. Borel [18]: *... had to accept the continuum as a primitive concept, not reducible to an arithmetical theory of the continuum [numbers as points, continuum as a set of points].*
- L. Brouwer [18]: *The continuum as a whole was intuitively given to us; a construction of the continuum, an act which would create all its parts as individualized by the mathematical intuition is unthinkable and impossible. The mathematical intuition is not capable of creating other than countable quantities in an individualized way. [...] the natural numbers and the continuum as two aspects of a single intuition (the primeval intuition).*

In the mid-1950s there were some attempts to comprehend the intuitive notion of Continuum by giving it strictly computational and constructive sense, i.e. by considering computable real numbers and computable functions on those numbers, see Grzegorczyk[7] [8] [9], and Lacombe [14]. These approaches were mainly logical and did not find a ubiquitous interest in Mathematics.

If the concept of Continuum is different than the concept of number, then the problem of reconstructing the computational grounding for the Continuum is important.

For more interesting and important discussion, with a historical background, see [5, 2, 15].

Recently, see HoTT [19], a type theory was introduced to homotopy theory in order to add computational and constructive aspects. However, it is based on Martin Löf's type theory that still is a formal theory invented to provide intuitionist foundations for Mathematics. The authors of HoTT admit that there is still no computational grounding for HoTT.

Robert Harper [10] : *"... And yet, for all of its promise, what HoTT currently lacks is a computational interpretation! What, exactly, does it mean to compute with higher-dimensional objects? ... type theory is and always has been a theory of computation on which the entire edifice of mathematics ought to be built. ... "*

The Continuum is defined in HoTT in the usual way as the real numbers via Cauchy sequences.

Since the intuitive notion of Continuum is common for all humans (not only mathematicians), the computational grounding of the Continuum (as a primitive type) must be simple and obvious.

The proposed grounding is extremely simple, and may be seen as naive. It corresponds to the Brouwer notion of Continuum, see Section 8.

The introduced new primitive types along with constructors, primitive operations and primitive relations should be seen as a part of the general framework for a constructive type theory presented in the work Types and operations (for short TO) see [1] `http://arxiv.org/abs/1501.03043`.

1.1 Informal introduction of Continuum

According to the American Heritage®Dictionary of the English Language: *"Continuum is a continuous extent, succession, or whole, no part of which can be distinguished from neighboring parts except by arbitrary division"*.

Intuitively continuum (as an object) can be divided finitely many times, so that the resulting parts are of the same type as the original continuum. Two adjacent parts can be united and the result is of the same type as the original continuum. Based on this simple intuition, continuum may be interpreted as an analog signal transmitted in a link.

Here the link corresponds to the type Continuum whereas a signal corresponds to an object of the type Continuum. In telecommunication there are natural examples for link division, like frequency division and time division, as well as for link merging especially in the optical networks.

While dividing a link into two (or more) parts it may happen that in some parts there are no signals. These parts are deactivated (blinded) and are not taken to next divisions.

As a result of divisions and deactivations, a structure of active parts is emerged where each part is of the type Continuum, some parts are adjacent, and the structure as a whole is also of the type Continuum. This very structure may be interpreted as an approximation of the signal in the link, more accurate as the divisions are finer.

If interpreting Continuum as a physical link, there are many kinds of Continuum, of different dimensions, of different methods of division, as well as of many criteria for deactivation of blind parts of a link. However, there are common features that constitute together the essence of the Continuum. This very essence is formalized below.

2 Cubical complexes and Continuum

Still as an intuitive example.

Since the closed unit interval $[0, 1]$ (a subset of real numbers $\mathbb{R}$) is a mathematical example of continuum, let us follow this interpretation, however with some restrictions. Let the n-th dimensional continuum be interpreted as the unit n-th dimensional cube, that is, the Cartesian product of n-copies of the unit interval.

Let us fix the dimension and consider a unit cube. The cube may be divided into parts (smaller cubes) in many ways. However, the most uniform division is to divide it into 2^n the same parts, where n is the dimension of the cube. Each of the parts may be divided again into 2^n same sub-parts (smaller elementary cubes), and so on, finitely many times. Some of the parts may be removed as they are interpreted as empty (blind) in the link. The resulting structure is exactly a uniform cubical complex (see for example [13]) consisting of elementary cubes all of them of the same dimension. There is a natural relation of adjacency between the elementary cubes. Two cubes (parts) are adjacent if their intersection (as sets) is a cube of dimension $n-1$. Although the complexes can be also simplicial or hexagonal, it seems that the cubical ones are most natural at least in Mathematics.

Let a uniform cubical complex (complex, for short) be denoted by e. Two adjacent parts of e may be united into one part. After uniting some adjacent parts, the resulting structure is called a manifold by the analogy to manifolds in algebraic topology. See Fig. 1 for an example of 2-dimensional unit cube that is divided, and then some of the parts removed, and adjacent parts are united.

A complex may be arbitrary large, i.e. may contain a large number of parts, and a sophisticated adjacency relation, however, usually its essential structure (information it contains) is relatively small and does not depend directly on the number of the parts and of the adjacency complexity. For this very reason the uniting, preserving this essential structure and reducing significantly the number of parts, is of great importance.

Denote by $\hat{e}$ the manifold resulting from uniting the all adjacent parts in e.

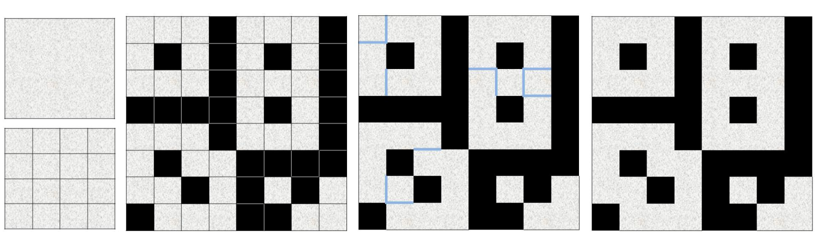


Figure 1: Divisions, deactivations, and uniting

The final manifold $\hat{e}$ consists of disconnected components (segments), see Fig. 1, the first example from the right.

For complex e, let its complement (consisting of empty black cubes (parts)) be denoted by e^{-1}. It is dual to e and it is also the subject to uniting its parts. The white final manifold $\hat{e}$, and its dual black final manifold $\hat{e}^{-1}$ contain some essential information of the original object, i.e. the complex e. The adjacency relation between the white segments

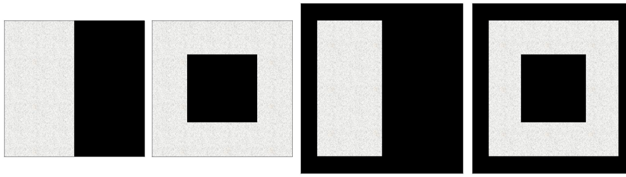


Figure 2: Examples of $\hat{e}$ and $\hat{e}^{-1}$ without black border and with black border

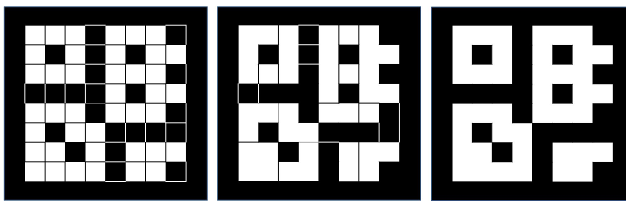


Figure 3: Objects with one black outer border

of $\hat{e}$ and the black segments of $\hat{e}^{-1}$ reflects to some extent the structure of e and e^{-1}. However, this relation is not complete, that is, it can not distinguish between two different cases shown in Fig. 2, see the first two examples. Note that the adjacency between the white segments (as well as black segments) to the border of the original unit cube is important and can not be neglected. The adjacency to the border can be eliminated if the border is fixed as the one black segment, see Fig. 2 and Fig. 3. This simplifies (by aggregation) the adjacency relation between black and white segments. Denote this simplified aggregated relation by R_e.

From now on, let the original unit cube have the black border as the outer segment.

Note that the relation R_e concerns complex e as well as its complement e^{-1}. For this very reason the complex must be considered as a collection of small elementary cubes within the unit cube with the black border. The complement of a complex is taken relatively to this very unit cube. Hence (e, e^{-1}) is the right object to be considered. We will use only single symbol e to denote a complex , however, with the context of its complement e^{-1} in the unit cube with the black border.

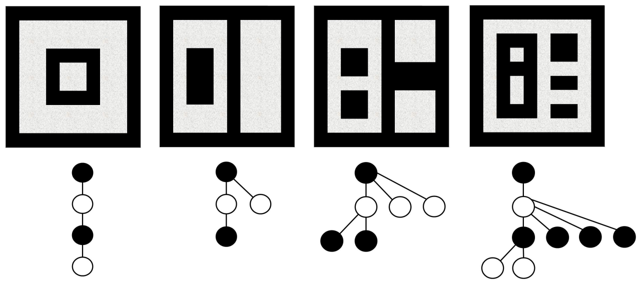


Figure 4: More final dual manifolds and the corresponding trees

Note that the number of white segments and the number of black segments alone are not sufficient to express important information of a complex, see Fig. 4 and the two first examples where the number of white segments and the number of black segments are the same.

The aggregated adjacency relation R_e contains more essential information of the object (e, e^{-1}). It gives rise to define the following equivalence relation for complexes. Two complexes e_1 and e_2 are *similar* if the relations R_{e_1} and R_{e_2} are isomorphic.

More interesting relations may be introduced by using homomorphisms between relations R_{e_1} and R_{e_2}.

Any black segment (except the one that contains the border) is externally adjacent exactly to one white segment. Also any white segment is externally adjacent exactly to one black segment. Hence, R_e can be represented as the following tree. The `root` of the tree is the black segment containing the border. The `node`s of depth 1 (children of the root) are the white segments adjacent (according to R_e) to the black root. Each of the white `node` of depth 1 may have inner adjacent black segments as its children (`node`s of depth 2). Each of the black `node`s of depth 2 may have inner adjacent white segments as its children (`node`s of depth 3). And so on. See Fig. 4 for examples for complexes of dimension 2. However, for higher dimensions the relation R_e is not sufficient to distinguish different manifolds, like tori and sphere in dimension 3, i.e they have the same simple tree (*black* $\rightarrow$ *white* $\rightarrow$ *black*) corresponding to their adjacency relations between black and white segments. More simple example consists of a solid tori, and a ball in dimension 3. Hence, for complexes of dimension 3 and higher the aggregated adjacency relation R_e reflects only partial information contained in these complexes.

Note that R_e as black-white tree may be considered as primeval *shape* of the complex e.

It is clear that there must be different and more subtle uniting criteria to reduce the number of parts, and to aggregate the adjacency relation for white complex as well as for its dual black complex. Final result of such uniting should represent important information contained in complexes. The problem is to find out the criterion that gives the all essential information. The term *essential* should correspond to homotopy theory.

Conjecture. Two complexes (as topological subspaces of two dimensional Euclidean space $\mathbb{R}^2$) are of the same homotopy type iff they are similar.

There are more complex uniting criteria, see [12] for reduction algorithms to compute homology of finitely generated chain complexes. Since we are interested in fundamental principles, the uniting criterion must be simple. It seems that the most simple and natural one is presented above. It should be sufficient for 2-dimensional complexes.

Note that considering complex e along with it complement e^{-1} as *the one structure* is the main difference between classic notions of complexes and the one introduced above. This enforces the dimension to be fixed, i.e. complexes are uniform and consist of cubes of the same dimension. Also the complement, the black cubes are essential although they may be viewed as nothing (empty space). Together with the white cubes they form the original initial unit cube.

Intuitively, the computational and constructive grounding of the notions presented above are clear and obvious. It is not true. There are interpretations of these notions in combinatorial structures that abstract from intuitions of the Euclidean space. However, simplicial (cubical sets) and categories are more advanced abstractions where Euclidean space is regarded as the space for geometric realizations. Hence, these interpretations lead from one abstraction (Euclidean) to a more complex and higher abstraction (categories), where computational grounding is even harder to understand.

The investigations presented above still contain implicit intuitions corresponding to the Euclidean space. They are simple and typical for usual mathematical reasoning.

In the following sections an attempt is presented to reconstruct the generic and complete computational and constructive grounding, for which the introduced above notions of manifold and the similarity relation are special cases.

3 A generalization

The purpose is to liberate (step by step) our reasoning from the Euclidean space interpretation.

Note that in the previous section only the following properties were essential.

- Any part is the same as the original unit cube, i.e. it can be divided arbitrary many times into smaller parts, and these smaller parts are the same as the unit cube. The pattern of the adjacency relations between these parts is repeated in the subparts during the consecutive divisions.
- Any part (except the border parts) has the same number of its adjacent parts and the same adjacency structure.

Hence, what matters is:

- the adjacency relation between parts;
- the fractal like structure (self-similarity) of unit cube, i.e. any part is the same as the original cube. This self-similarity concerns also the adjacency relation.

For this reason, let us abstract the unit cube and the uniform cubical complexes from Euclidean space by considered only these properties.

3.1 A generic framework: introducing a new primitive type corresponding to the Continuum

The investigations presented below should be viewed as an attempt to grasp the computational and constructive grounding of Continuum.

Let us again cite Robert Harper [10] *"... type theory is and always has been a theory of computation on which the entire edifice of mathematics ought to be built. ... "*

What are types? There are primitive types, and there are types constructed from the primitive ones by using constructors, like product, disjoin union, arrow, dependent type constructors Σ and Π, and much more.

Introducing a new primitive type requires: primitive objects, object constructors and destructors, and complete elementary relations for this type (as parametrized primitive types). One of the relations is equality. Complete elementary relations means (see [1]) that negation is well founded (grounded). The type of natural numbers N was introduced (see [1]) in the very similar way.

Let the abstract unit be denoted by C_0. It is to be divided according to a fixed partition pattern consisting of subparts and the adjacency relation between them. In the consecutive divisions each subpart (sub-sub-part, and so on) can be divided according to the same pattern.

Let the *p-pattern* (partition pattern) be denoted by (C_1, Adj_d) where C_1 is a finite collection (type) of parts of the unit C_0, and Adj_p is the adjacency relation between these parts. To be a partition of the unit, the parts must be connected, that is, uniting the all adjacent parts results again in the unit C_0. Equality relation on C_1 is supposed to be primitive and is denoted by Eq_{C_1}.

Let $\hat{C}_0$ denote the type having only one object C_0. It is convenient to extend Eq_{C_1} to $C_1 + \hat{C}_0$, that is, to have the equality relation on elements of C_1 with one additional element C_0. We abuse a bit the notation, so that C_0 will denote either $\hat{C}^0$ or the unit C_0; it will be clear from the context.

Let the number of the elements of C_1 be denoted by l, so that C_1 consists of the following elements $a_1,\ a_2,\ ...,\ a_l$. In the very convenient dot notation, the subparts (as result of the consecutive divisions) of C_0

are denoted as, for example, $a_{i_1}.a_{i_2}$ for the second division, and $a_{i_1}.a_{i_2}.a_{i_3}$ for the third division, and so.

Let us introduce a new primitive type denoted by C. It is an example of Martin-Löf's W-type, known as the types of well-founded trees. However, here C is equipped with adjacency relation.

The abstract unit C_0 is the primitive object of type C. The constructors are as follows.

For any a (belonging to C_1) there is a successor operation $Succ_a : C \rightarrow C$ such that for any $c : C$, $Succ_a(c)$ (also denoted by $c.a$) is object of C.

There is one destructor $Pred_C : C \rightarrow C$, such that for any $c.a$ of type C, $Pred_C(c.a)$ is c. Let $Pred_C(C_0)$ be C_0.

The objects of type C may be interpreted as stacks in programming with fixed initial (bottom) element C_0 .

Getting the top element of a stack is primitive operation $Top_C : C \rightarrow (C_1 + C_0)$, such that $Top_C(C_0)$ is C_0, and $Top_C(c.a)$ is a.

Actually the type C has the structure of a tree with objects as finite branches starting with the root C_0. It may be interpreted as a type of stacks where elements of C_1 can be pushed on and popped from. That is, push correspond to $Succ_a$, and pop to $Pred_C$ and Top_C together.

Let the primitive operation $length_C : C \rightarrow N$ be introduced such that $length_C(C_0)$ is 1, and $length_C(c.a)$ is $length_C(c) + 1$. Although it is an inductive definition, it it not a construction. It may be constructed from more primitive notions, however, then these very notions must to be introduced first as primitive ones. It seems that rather $length_C$ should be primitive, and seen as a built-in attribute of object of type C during this object construction.

Let the collection of objects of C of length $k+1$ be denoted by C_k. Note that it is a property for a fixed k that can be easily constructed. We will abuse the notation because in this way an object of C_1 has double connotation, once as part a of the unit C_0, and second as $C_0.a$. It will be clear from the context what we mean by C_1.

Complete elementary relations (to be constructed) for type C are: $Equal_C$, $Ancestor_C(c_1; c_2)$ (i.e. c_2 is an initial segment of c_1), $Predecessor_C(c_1; c_2)$ (i.e. c_1 is an initial segment of c_2). Their complement is $Incomparable_C$. They can be constructed from the primitives presented above. Actually, $Predecessor_C + Equal_C$ is a partial order on C, and is denoted by $\sqsubseteq_C$.

The relations are mutually disjoint and complete, that is, the negation (complement) of any of them is the disjunction (disjoint union) of the rest. They are operations of type $(C; C) \rightarrow Types^1$.

There are another complete elementary relations that are introduced in the next Section 4.

Construction of $Equal_C$ is presented below.

First, let's construct the operation $qui : C \rightarrow (N \rightarrow (C_1 + C_0))$ where $qui(c)$ (for $c : C$) is the sequence of the consecutive elements pop-ed from the stack c. That is, if c is $C_0.a_1.a_2....a_{k-1}$, then $qui(c)(1)$ is a_{k-1}, and $qui(c)(2)$ is a_{k-2}, ... $qui(c)(k-1)$ is a_1, and $qui(c)(k)$ is C_0. For n greater than k, $qui(c)(n)$ is C_0.

The construction of qui is as follows. Let E denote the type $N \rightarrow A$, and A denote $(C_0 + C_1)$.

For the construction of the auxiliary operation $op : (C; E; N) \rightarrow (C; E; N)$ see the first (from the left) construction of Fig. 5.

First, the composition of $Copy_C$ and $Pred_C$ and $Proj_C$ is done. It has two outputs: one of type C and the other of type A.

Recall (see TO [1]) that $Change_A : (A; N; (N \rightarrow A)) \rightarrow (N \rightarrow A)$ is a primitive operation such that $Change_A(a; n; f)$ is the operation g such that g is the same as f except input n where $g(n)$ is a.

Since the input type and output types of op are the same, op can be iterated, See TO [1] for primitive operation $Iter_D$.

Let D denote the type $(C; E; N)$. Then $Iter_D(k; op)$ is the k-th iteration of operation op. Let $q : N \rightarrow C$ be the constant operation with output C_0 for all $n : N$. Then, to construct qui see the Fig. 5.

The construction results in operation of type $C \rightarrow (C; E; N)$ having triple output. Neglecting the outputs C and N, we get the required operation qui.

$Equal_C(c_1; c_2)$ is constructed as the condition:
$Equal_N(length_C(c_1); length_C(c_2))$ and for any $i = 1, 2, \ldots, length_C(c_1)$:
$Eq_{C_1}(qui(c_1)(i); qui(c_2)(i))$.

For generic constructors of conditions see TO [1].

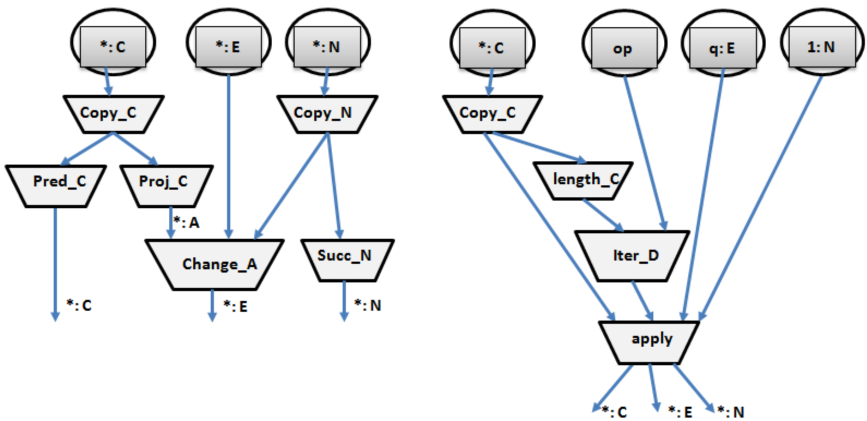


Figure 5: Construction of operation *op* and *qui*

The rest of the complete elementary relations (i.e. $Ancestor_C$, $Predecessor_C$), can be constructed in a similar way.

Note that so far the adjacency relation Adj_p was not used, so that the type C is merely a Martin-Löf's W-type.

4 Adjacency relation

In order to liberate our reasoning from the intuition of Euclidean space, it is convenient to consider the adjacency relation Adj_p on C_1 as a graph where edge between two vertices corresponds to the adjacency between them.

The essential aspect of C is the adjacency relations between its objects of the same length, say k, i.e. objects of C_k. For a fixed c, the adjacency between objects $c.a_1$, $c.a_2$, ..., $c.a_l$ (where a_1, a_2, ..., a_l are elements of C_1) is determined by the p-pattern. Recall that it is a finite partition C_1 of the original unit C_0 along with the adjacency relation Adj_p on C_1 such that the corresponding graph is connected, i.e. to be a partition of the unit, uniting the all adjacent parts results in one component that is C_0.

To extend the adjacency relation onto C_k, a merging pattern (*m*-

pattern for short) is needed. In the case of cubic divisions and their interpretation in Euclidean space, the m-pattern is obvious.

In a generic form this m-pattern is defined as follows. Two adjacent parts a_i and a_j (i.e. $Adj_p(a_i, a_j)$ in the p-pattern) determine which objects $a_i.a_k$ and $a_j.a_n$ are adjacent, where a_k and a_n are elements of C_1. Let this adjacency relation be denoted by Adj_m. Then, $Adj_m((a_i.a_k);(a_j.a_n))$ means that $Adj_p(a_i, a_j)$ and $a_i.a_k$ and $a_j.a_n$ are adjacent according to the m-pattern. Actually Adj_m is a relation (graph) on C_2.

IMPORTANT condition: The graph corresponding to m-pattern must be connected. For the partitioned unit to be connected (it must remain a unit after joining all adjacent parts), for any two adjacent parts a_i and a_j (i.e. $Adj_p(a_i; a_j)$, there are a_k and a_n such that $Adj_m((a_i.a_k);(a_j.a_n))$.

In what follows the relation $Adj(k; c_1; c_2)$ (for $k : N$, and c_1 and c_2 of type C) is constructed in an informal way referring to the construction methods introduced in TO [1]. It is assumed that p-pattern and m-pattern (as Adj_p and Adj_m) are fixed.

Intuitively, two objects c_1 and c_2 of the same length are adjacent if all their prefixes of the same length are either equal or adjacent.

For c_1 and c_2 of type C, and of length $k + 1$, let i be the least number such that prefixes of length $k + 1 - i$ of c_1 and c_2 are equal. The construction of such i is as follows. Let the auxiliary operation $o : (C; C; N) \to (C; C; N)$ be constructed so that $o(c_1; c_2; n)$ consists of the three outputs: $Pred_C(c_1)$, $Pred_C(c_2)$ and $Succ_N(n)$.

The primitive operation *while* (see TO [1], Section 10.1.1.) with condition $\neg Equal_C$ and operation o, gives operation that for the input k (iterate k-times the operation if the condition is true), c_1 and c_2 (both of the length $k+1$), and input 1; gives the output $c_{1'}$ and $c_{2'}$ such that $c_{1'}$ and $c_{2'}$ are maximal equal prefixes of respectively c_1 and c_2, and the output of type N is the required i. Note that i is less than $k+1$, because all objects of C may be interpreted as branches in the tree with the root C_0.

Given the construction of i, $Adj(k; c_1; c_2)$ is constructed as the conjunction of the following conditions.

- $Equal_N(length_C(c_1); k+1))$ and $Equal_N(length_C(c_2); k+1))$, i.e. c_1 and c_2 are of the same length $k+1$.

- Since $c_{1'}$ and $c_{2'}$ are maximal equal prefixes and are of length $k+1-i$, for $c_{1'}.qui(c_1)(i)$ and $c_{2'}.qui(c_2)(i)$ to be adjacent it is necessary that
  $Adj_p(qui(c_1)(i); qui(c_2)(i))$.

- For $c_{1'}.qui(c_1)(i).qui(i-1)$ and $c_{2'}.qui(c_2)(i).qui(i-1)$ to be adjacent it is necessary that
  $Adj_m(qui(c_1)(i).qui(c_1)(i-1); qui(c_2)(i).qui(c_2)(i-1))$.

- For for $c_{1'}.qui(c_1)(i).qui(i-1).qui(i-2)$ and $c_{2'}.qui(c_2)(i).qui(i-1).qui(i-2)$ to be adjacent it is necessary that the above conditions is true and
  $Adj_m(qui(c_1)(i-1).qui(c_1)(i-2); qui(c_2)(i-1).qui(c_2)(i-2))$.

- and so on

- $Adj_m(qui(c_1)(3).qui(c_1)(2); qui(c_2)(3).qui(c_2)(2))$.

- And finally $Adj_m(qui(c_1)(2).qui(c_1)(1); qui(c_2)(2).qui(c_2)(1))$.

This completes (somehow informal) construction of the relation $Adj(k; c_1; c_2)$. Note that $\neg Adj$ can be constructed on the basis of the negations of Adj_p and Adj_m . Hence, Adj and $\neg Adj$ has been constructed as complete relations of type $(N; C; C) \rightarrow Types^1$.

Actually, the adjacency relation Adj is restricted to the objects of the same length, so that it will be considered separately for any k, and if C is fixed, it will be denoted by Adj^k.

However, the requirement that the relation Adj_m is a connected graph is not enough. In order to conform to the intuition of the Continuum, for any $k : N$, the relation Adj^k must be a connected graph. See the next Section and Fig. 6.

Although for the simplicity of the presentation, the descriptions of the above constructions are informal, the explicit constructions must be provided for the completeness of the proposed framework, i..e. the constructible Universe proposed in TO [1].

The number of objects of C_k (of length $k+1$) is $|C_1|^k$, where $|C_1|$ is the number of elements of C_1. In order to construct an operation that enumerates the objects of C_k, i.e. $enum_C(k; *) : N' \to C$ let $enum_C(1; *) : N' \to C$ denotes a fixed linear order on C_1. For any $k : N$, the enumeration $enum_C(k; *) : N' \to C$ is the lexicographical order on C_k determined by the order on C_1. To be exact, the output $enum(k; n)$ for any n greater than $|C_1|^k$ is fix to be C_0. The enumeration can be constructed. Let us assume that it is done.

To conclude this section. Any two patterns (p-pattern and a corresponding m-patter) give rise to introduce a primitive type denoted by C. The relation Adj^k is inherent part of the construction of C. Note that this type and the relation should be indexed by these very patterns, so that actually a primitive operation is introduced that may be viewed as a generic primitive type constructor that for any p-pattern and m-pattern gives a new primitive type C along with its adjacency relation Adj^k.

Note that the fractal like structure of C_k is the result of its construction via p-pattern and m-pattern.

5 Patterns of Continuum

According to the ubiquitous intuition of the Continuum, *" ... no part of which can be distinguished from neighboring parts except by arbitrary division"*.

Hence, the following properties are essential.

- *Indiscernibility property:* Any permutation of the parts of C_1 is an isomorphism relative to the p-pattern adjacency relation Adj_p and m-pattern adjacency relation Adj_m. (No part is distinguished.) Then, in p-pattern, for any part the number of its adjacent parts is one and the same and is defined as the *dimension*.

- *Homogeneity property:* For any c of type C and any part a_1 of C_1, the collection of all a_2 of C_1 such that $c.a_1$ and $c.a_2$ are adjacent (i.e. $Adj^k(c.a_1; c.a_2)$) is independent of the choice of c except the case it is a border object (to be defined below). More restricted condition is that such collections are isomorphic relatively to the adjacency relation Adj^k.

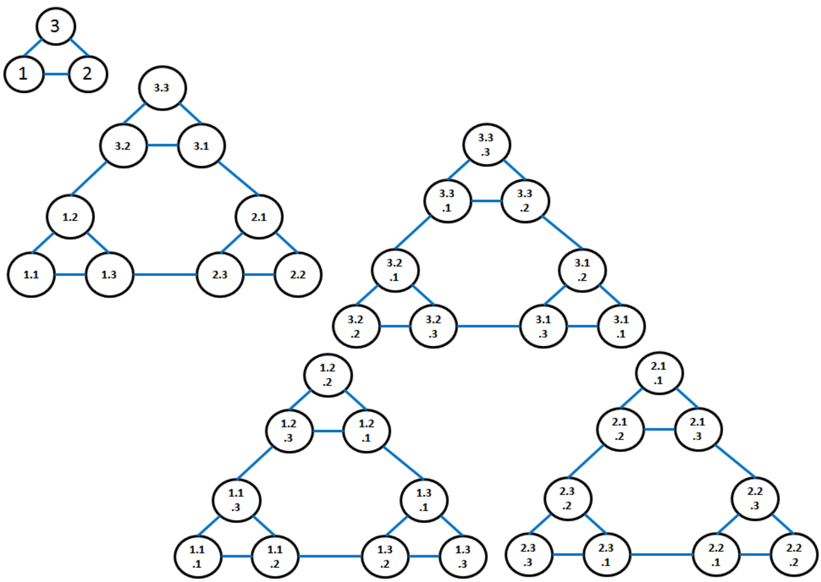


Figure 6: Simple p-pattern, m-pattern, and C_2

Adjacency relation Adj^k corresponds in the obvious way to the notion of finite topology on C_k, as special case of Alexandroff topologies where each point (object of C_k) has the smallest neighborhood consisting of itself and its adjacent objects.

In what follows the elements of C_1 are denoted by natural numbers 1,2, ... , n, where n is the number of elements of C_1.

Let us consider a simple p-pattern consisting of three elements 1,2, 3 of C_1 such that $Adj_p(1;2)$, $Adj_p(1;3)$, and $Adj_p(3;2)$.

An example of a simple m-pattern is following. Note that the adjacency relation is symmetric.

- the following objects are m-adjacent:
  1.3 and 2.3

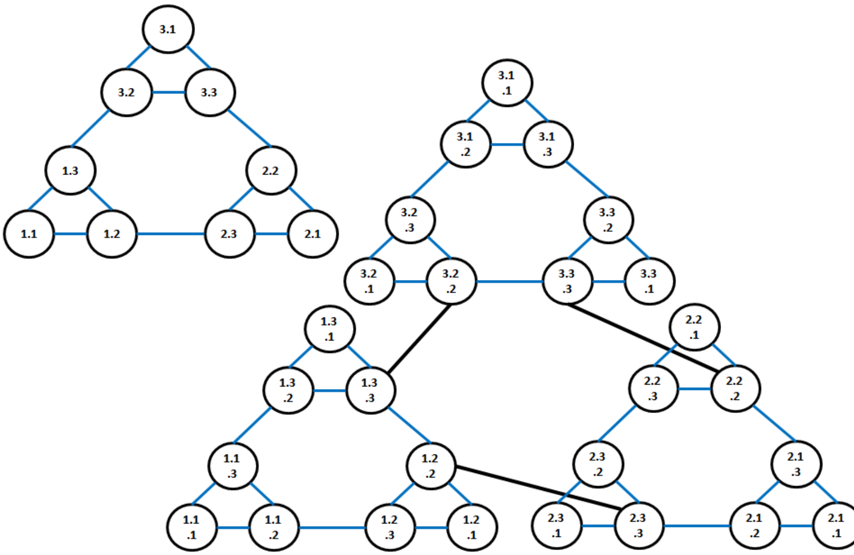


Figure 7: More complex m-pattern

2.1 and 3.1
3.2 and 1.2

The above m-patter does not satisfy the homogeneity property. The graph Adj^2 is not connected (see Fig. 6). It is counterintuitive because it is a division of the unit, so that after uniting of its all adjacent parts it should be also one unit.

However, for the following m-pattern. the graph Adj^2 is connected, see Fig. 7.

- the following objects are m-adjacent:
  1.3 and 3.2
  2.2 and 3.3
  2.3 and 1.2

The above m-patter also does not satisfy the homogeneity property,

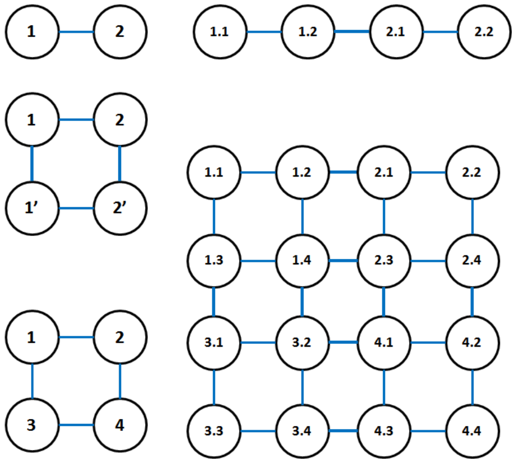


Figure 8: Euclidean patterns

5.1 Euclidean patterns

The indiscernibility and homogeneity properties, corresponding to the intuitive notion of Continuum, seem to be completely abstracted from any concrete realizations. However, they are related to Euclidean spaces, that is, the p-patterns and corresponding m-patterns presented below satisfy the indiscernibility and homogeneity properties.

The simplest Euclidean p-pattern for C_1 consisting of two elements 1 and 2 that are adjacent. There are several corresponding m-patterns, however, only one satisfies the homogeneity property. See Fig. 8. These patterns correspond to 1 dimensional Euclidean space, that is identified with the real numbers $\mathbb{R}$. However, still a lot of constructions must be done to get the constructive content of the notion of the real numbers.

The 2-dimensional Euclidean patterns are constructed as follows. Copy 1-dimensional Euclidean d-pattern and join the original with the copy. The same (original and its copy) parts are adjacent. It is shown in Fig. 8. There are four parts 1,2,3 and 4 . Parts 1 and 3 are not adjacent, and 2 and 4 are not adjacent in the p-pattern. Also in this case there is only one m-pattern that satisfies the homogeneity property, see Fig. 8.

For 3-dimensional Euclidean patterns, the method applied to 2-

dimensional case is repeated. That is, the 2-dimensional p-pattern is copied. Then, the original and the copy is joined into the new 3-dimensional p-pattern where the same (original and its copy) parts are adjacent. Analogously the corresponding m-pattern is constructed.

The same method may be applied to construct Euclidean patterns of any dimension, i.e. copy p-pattern and join into the higher dimensional p-pattern where the same parts are adjacent.

Note that the method for constructing patterns of higher dimensional Euclidean patterns corresponds to the Cartesian product. The essence of the Cartesian product is copy and join.

Let C^1 denote the the type C constructed for the Euclidean patterns of dimension 1. It corresponds to the unit interval of the real numbers, see Section 8. Generally, let C^n denote the type C constructed for the n-dimensional Euclidean patterns. It corresponds to the n-th dimensional cube.

It is interesting if there are non Euclidean patterns that satisfy the indiscernibility and homogeneity properties.

5.2 Border objects

Let the patterns satisfy the indiscernibility property. Dimension is fixed as n, i.e. it means that any element of C_1 has exactly n adjacent elements.

Any object $c : C$ of length $k+1$ may be denoted, in the dot notations, as $C_0.qui(c)(k).qui(c)(k-1).\ (\dots)\ .qui(c)(2).qui(c)(1)$

Border objects of C_k are defined as follows.

- 0-border objects (corners): c of C_k is 0-border object if for $j = 1, 2, \dots, k$, all $qui(c)(j)$ are the same.
- 1-border objects (edges): c of C_k is 1- border object if there are two adjacent elements a_1 and a_2 (belonging to C_1) such that $Adj_p(a_1; a_2)$ and for any $j = 1, 2, \dots, k$: $qui(c)(j)$ is either a_1 or a_2.
- 2-border objects (faces): c of C_k is 2- border object if there are three parts a_1, a_2 and a_3 (belonging to C_1) such that they are

connected in p-pattern, and for any $j = 1, 2, \ldots, k$: $qui(c)(j)$ is either a_1 or a_2 or a_3.

- objects of higher border order (less than n) are defined in an analogous way.

5.3 Summary of the generic approach

Starting with the notion of Euclidean space (that is considered as a concrete notion in classic Mathematics) abstraction was done to a primitive type. It was done by introducing abstract unit, divisions and parts of the unit, p-pattern and m-pattern, and optionally the properties of indiscernibility and homogeneity. Number of divisions is potentially unlimited. That's all.

The result is a primitive inductive W-type denoted by C along with the primitive operation $length_C$. Complete relations were constructed: $Equal_C$, $Ancestor_C$, $Predecessor_C$, and $Incomparable_C$. This is not novel. However, the adjacency relation Adj^k (also its complement $\neg Adj^k$) constructed from p-pattern and m-pattern as inherent component of the primitive type C, seems to be novel. Within the framework TO [1], these complete relations may be used to construct sophisticated structures corresponding to mathematical notions.

See Section 8 for discussion how the type C does correspond to the notion of the set of real numbers.

6 Coloring, abstract complexes, and pre-shapes

The constructions described (informally) below resulted from an attempt towards (re)constructing the computational content of the notion of homotopy types.

Let's introduce the border b as the primitive and the only one object of the singleton type B. Let $C + B$ be denoted by $\bar{C}$. Let $\bar{C}_k$ denote C_k with the border.

Extension of the adjacency relation Adj^k to $\bar{C}$ can be constructed; i.e. $\bar{Adj}^k(c, b)$ holds iff c is a border object of C_k.

Operations of type $e : \bar{C} \to N$ is called *abstract complex* if it has only two output objects 1 for white, and 2 for black. This may be generalized to more colors. Let e^k denote the restriction of e to $\bar{C}_k$.

At last we have arrived in the abstract, general and *constructive* (i.e. having clear computational content) method for representing objects for which the uniform cubical complexes (together with complements) from Section 2 are special cases. Now, the informal and non constructive (however, strictly mathematical) reasoning, from Section 2, is presented below as consecutive constructions. For the clarity of the presentation, the constructions are described informally, however, with sufficient details they are to be implemented within the framework of TO [1].

The relation $\bar{Adj}^k$ can be seen separately for black objects as the relation $\bar{Adj}^k(c_1; c_2) \times Equal_N(e(c_1); 2) \times Equal_N(e(c_2); 2)$ (denoted by $\bar{Adj}^k_b(c_1; c_2)$), and for white objects as the relation $\bar{Adj}^k(c_1; c_2) \times Equal_N(e(c_1); 1) \times Equal_N(e(c_2); 1)$ denoted by $\bar{Adj}^k_w(c_1; c_2)$.

Note that the transitive closure of $\bar{Adj}^k_b$ (and $\bar{Adj}^k_w$) is an equivalence relation. The equivalence classes correspond to the color connected components of the complex. This transitive closure is constructed below. Although the construction is simple, it is presented in order to discuss the computational grounding of *quotient types* in type theory.

For the color i, and for any $k : N$ and c_1 and c_2 of $\bar{C}_k$ (of the same color i, i.e. $Equal_N(e(c_1); i)$ and $Equal_N(e(c_2); i)$. A path from c_1 to c_2 is defined as an operation (sequence) $path_{(c_1,c_2)} : N \to \bar{C}$ constructed below.

Initially, $path_{(c_1,c_2)}$ is set as the constant operation $const_{(N,C)}(C_0)$, i.e. with the constant output C_0. Then, the first object of the path is set to be c_1, i.e. $Change_C(c_1; 1; path_{(c_1,c_2)})$.

The path construction consists in incremental expansion of the path starting with c_1 to reach the destination object c_2. The number of objects of C_k is $|C_1|^k$, and there is already constructed enumeration operation of C, that is, $enum_C$, so that its extension to $\bar{C}$ is simple and denoted by $enum_{\bar{C}} : (N; N') \to \bar{C}$. The incremental expansion is done iteratively according to the following condition.

- Take the next object of C_k in the enumeration of C_k. If it is c_2, then the path is already constructed. Otherwise, if it is of color i, and is adjacent to an element of the already constructed path, then join it to the path.

Finally, two objects c_1 and c_2 of $\bar{C}_k$ of the same color (i.e. $Equal_N(e(c_1); i)$ and $Equal_N(e(c_2); i)$), are equivalent if there is a path between them.

Note that this corresponds to the paths in homotopy theory. However, the above paths are neither continuous functions in homotopy theory, nor objects of the equality type $Id_C(c_1, c_2)$ in HoTT [19]. Actually, the investigations presented here are supposed to be primitive, and to expose the constructive aspects of the more abstract notions.

For a complex e^k, a segment is defined as all equivalent objects (connected) of the same color. The equivalence classes correspond to the segments of complex e^k, and together may be considered as the so called *quotient type*. For the segments to be objects of this *quotient type*, a generic primitive constructor for such types and objects should be introduced. It seems that it is impossible, so that the notion of *quotient type* is not strictly type theoretic (with pure computational and constructive content); it is an abstract notion. There is a construction from which the abstract notions of the *quotient type* can be derived.

In the following construction the abstract notions (i.e. color segments), and the aggregated adjacency relation between the segments can be interpreted, that is, can have grounding.

For a fixed e, let us construct the operation q_e of type $(N; \bar{C}) \rightarrow (N; N')$, such that for c of $\bar{C}_k$, if $q_e(k; c)$ is $(n; i)$ (where i is the same as $e(c)$, and n is greater than 1), then c belongs to the n-the segment of color i. The output $(1; i)$ is reserved for $q_e(k; c)$ such that c is not in $\bar{C}^k$ and its color is i.

The construction of q_e is based on the above path construction, $enum_{\bar{C}}$, and the fact that the number of objects of $\bar{C}_k$ is determined in advance and it is $|C_1|^k$.

Hence, each segment of e^k is represented by its color and unique number starting with the number 2.

Once the operation q_e is constructed, the equivalence relation is also determined, that is, two objects c_1, c_2 of $\bar{C}_k$ are in this relations if they belong to the same segment, that is, if $q_e(k; c_1)$ is the same as $q_e(k; c_2)$.

Hence, a segment of complex e^k, and of color i, is identified with $(n; i)$ such that there is c such that $q_e(k; c)$ is $(n; i)$. Let such $(n; i)$ be called a k-segment of the complex e. Enumeration of the segments of complex e^k can be constructed.

In the notation of Section 2, the aggregated adjacency relation and the relation R_e between two k-segments of complex e of different colors, can be constructed on the basis of the following condition.

Segments of different colors are represented by $(n_1; i)$ and $(n_2; j)$ where i is different than j.

- There are c_1 and c_2 of $\bar{C}_k$ such that $\bar{Adj}^k(c_1; c_2)$ and $q_e(k; c_1)$ is $(n_1; i)$, and $q_e(k; c_2)$ is $(n_2; j)$, and $\neg Equal_N(i; j)$.

Let the above relation be denoted again as R_e^k. It may be interpreted as a bipartite graph. However, the notion of bipartite graph as well as of tree are abstract ones, and so far have no constructive grounding. This issue will be discussed in the Section 6.2 below.

The relation R_e^k may be viewed as (partial) essential information that the complex e^k contains, and it may be called pre-shape.

Note that for Euclidean patterns (at least for dimensions 3 and less), the bipartite graphs of the abstract complexes are also black-white trees, see Section 2.

6.1 Sections

Although the relation R_e^k (where e is a complex and $k : N$) does not determine all essential information of the complexes e^k, it gives rise to next interesting constructions that are still elementary. Some of them are sections.

Consider C_1 with adjacency relation (the p-pattern) as a graph. Consider a partition of C_1 consisting of two (perhaps more) A_0 and A_1 that are connected subgraphs of C_1.

In order to construct the sections, the type of finite zero-one (0 and 1) sequences is needed. Denote this type by Sec, and by Sec^k the sequences of length k. Note that for this type analogous (as for the type C) operations may be constructed.

For $sec : Sec$, let $sec(i)$ denote the i-th element of sequence sec.

Assuming that 0 is less than 1, the lexicographical order can be constructed on Sec as operation (relation) Lex of type $(Sec; Sec) \rightarrow Types$. The order restricted to sequences of length k is denoted by Lex^k.

The operation $\sigma : C \rightarrow Sec$ can be constructed such that

for any c of C_k, $\sigma(c)$ is *sec* of length k such that for all $i = 1, 2, \ldots, k$, element $qui(c)(i)$ belongs to $A_{sec(i)}$.

Operation σ determines the sections in C_k for any $k : N$.

For any C_k, any section can be represented by an element of Sec^k.

If A_1 and A_2 have the same number of elements, then for Euclidean patterns of dimension n, any section is of dimension $n - 1$.

Example. For the Euclidean 2-dimensional pattern, if A_1 is a singleton consisting of only a_1, whereas A_2 consists of a_2, a_3 and a_4, then the corresponding sections look rather strange. However, if A_1 consists of a_1 and a_2, whereas A_2 consists of a_3 and a_4, then the corresponding sections are natural for any k.

Note that the sections correspond to the abstract notion of sheaves.

6.2 pre-Shapes as an abstract structure

Any complex e^k corresponds to the bipartite graph (V_{e^k}, E_{e^k}) where the set of vertices consists of black and white segments whereas the edges correspond to adjacency (i.e. R_e^k) between black and white segments.

An abstraction method (common in Mathematics) gives rise to the new abstract notion that may be called *abstract type*. In this particular example this abstract type refers to the bipartite graphs. Let it be called $pre - Shapes$.

Bipartite graphs are well known in Mathematics, however, here they are a new abstract notion. The canonical grounding of these $pre{-}Shapes$ are complexes e^k and corresponding relations R_e^k.

However, this abstract type may have different groundings. Note, that by grounding we mean a concrete structure that is explicitly constructed and has clear computational content.

Note also that this abstraction methods corresponds to the notion of *quotient types*, where equivalence classes are to be objects. This *quotient types* is an abstract type, and alone without a grounding has no computational content.

7 Abstractions

The primary goal of TO [1] and this paper is an attempt to reconstruct the grounding of some of the basic notions in Mathematics. It is supposed that this very grounding consists of all constructible objects (the Universe), that is, of parameterized finite structures that can be effectively (without abstractions) constructed and have clear unambiguous computational content.

Sets, subsets, functions, and equivalence classes are used ubiquitously in Mathematics. They are abstract notions. What is their grounding in the Universe?

The direction of the abstraction from Universe to the classic Mathematics seems to be right and corresponds to the order of the abstraction process when creating (learning) Mathematics. Sometimes this order is broken so that a student is given only a formal axiomatic approach to a mathematical theory without introducing him/her first to the so called intuitions, that is, the grounding from which (by abstractions) the theory in question is built.

Interesting and extremely important is the very process of abstraction. It seems that it is based on the general notions of set, function and predicate that correspond respectively to the notions of types, operations, and relations in the Universe.

7.1 Abstraction to a formal theory

Given a concrete construction as a collection of objects, operations and relations of some types, the abstraction process gets rids of the meaning (computational contents) of the objects, operations, relations and types leaving only symbols of types, symbols of objects, symbols of operations (function symbols), symbols of relations, and propositions as formal sentences. The abstraction process is not unambiguous (uniquely determined), it may concern only some of the object, operations, relations and types used in the construction in question. In particular it may lead only to bipartite graphs, or some other abstract algebra structures like group or field.

In general this kind of abstraction leads to predicate logic and specific formal theories of Gottlob Frege [6], that is, to a formal language with a signature for function symbols and relation symbols (predicates), and some sentences of the language as axioms of the theory. Logic provides derivation rules for formal reasoning (deduction). The axioms are supposed to be consistent together, that is, (semantically) no false sentence can be deduced from them, or (syntactically) no sentence and its negation can be derived. However, in the context of the abstraction process, and the fact that the formal theory is abstracted from a construction (being the prototype model of the theory), the theory is consistent.

Once a formal theory is fixed, then some other constructions may be abstracted to the same formal theory. The constructions may be seen as other models of the theory.

Once identifying the formal theory as an abstract type, these very structures (as models of this theory) are objects of this abstract type. This gives rise to consider homomorphism between them relative to that fixed abstract type. The isomorphisms, in turn, give rise to introduce equality for objects of this abstract type.

Note that here the problem of consistency of a formal theory (if the proper order of the abstraction process is preserved) does not exist. The very construction from which the formal theory is abstracted is a canonical model of this theory.

Note that homomorphisms between objects (models) of the same abstract type (formal theory) are concrete constructions in the Universe. This give rise to consider a new (abstract?) type of such homomorphisms (isomorphisms) for a fixed abstract type (formal theory).

7.2 Abstraction from potential infinity to actual infinity

This method of abstraction is ubiquitous in Mathematics. A concrete object (of some type) is either finite structure or constructed as a parametrized finite structure, i.e. finite structures indexed by natural numbers. So that for any concrete values of the parameters, the structure is finite, however, it may be arbitrary large, i.e. it is potentially infinite.

Abstracting from these parameters (or from some of them) results in an abstract object that contains the actual infinity.

In fact the notion of a real number (defined by Cauchy sequences) is a result of such abstraction.

7.3 Abstraction to sets

According to Errett Bishop [4], and Per Martin-Löf [16], to introduce a set is to define elements of the set as well as what does it mean that two of its elements are equal. Let's cite Bishop [4]:
To define a set we prescribe, at least implicitly, what we (the constructing intelligence) must do in order to construct an element of the set, and what we must do to show that two elements of the set are equal.

It seems that this captures the abstraction that leads to the notion of set. However, the view that a type can be defined as the set of all object of this type is, in general, senseless. For primitive types like the type of natural numbers and Continuum it makes some sense. However, definitely not for more complex types.

It seems that the notion of set is an abstraction of the notion of type and relation (property). The axiom of restricted comprehension states that for a given set X and a property of the elements of X there is the subset that consists exactly of the elements of X that have the property. This subset can not be grounded as a type. The notion of sub type is senseless. The grounding of the subset is just the property (relation) that is an operation, i.e. a concrete object.

Axiom of replacement states that image of a function is a set. It is abstraction of an operation, that is, the grounding of the image is the operation (being also the grounding of the function).

The axiom of choice is equivalent to the statement that every set can be well-ordered. This makes an illusion that for a given arbitrary set every of its elements can be constructed, or at least defined, listed. The transfinite induction allows to operate on the all elements of any set.

In Mathematic based on the set theoretic notation, the set theoretical (abstract) notions must have and actually do have grounding which is mostly implicit, and corresponds to the so called intuition.

7.4 Abstraction to functions

Let's again cite Bishop [4]:
A similar remark applies to the definition of a function: in order to define a function from a set A to a set B, we prescribe a finite routine which leads from an element of A to an element of B, and show that equal elements of A give rise to equal elements of B.

Hence, this abstraction presupposes the abstraction to sets. Then, abstraction is done from a concrete operation (as the very finite routine) by extending it to these sets.

In the following section these abstractions will be illustrated by reconstructing another grounding for the set of real numbers, and (continuous) functions on this set.

7.5 Abstraction to categories

Category is defined as a collection of objects and of arrows (also called morphisms) between objects. The arrows can be composed associatively, and there is an identity arrow for each object. It seem to be a high level abstraction of the (already abstract) notion of function.

This kind of abstraction corresponds also to the abstraction to formal theory. Once the group theory was abstracted from concrete constructions, the category of groups consists of all its models and homeomorphisms between them as arrows. The categories themselves can be treated as object (in category theory) and may form higher categories.

7.6 Definitions

A common view is that a definition is used to give a precise meaning to a new term. Definitions and axioms are the basis on which the Mathematics is described, and presented.

Usually definitions serve for denotation, that is, *a symbol, string of letters* denotes (as a shorthand) some complex term or relation expressed in a (formal) language. Such definitions are frequently used also in TO [1] and this paper to explain constructions of objects, operations, relations, and types. However, this kind of definitions (as shorthands) are for explanations, and are not so interesting.

The symbol (name of a concept) to be defined is called *definiendum*. *Definiens* is an expression in a formal language, usually in the language of set theory or category theory. If definition is not a denotation (a shorthand), the definiens is a formula (containing the symbol of definiendum) characterizing the notion being defined.

Given a collection (type, set, totality) of objects, and property of the objects, the objects that satisfy the property are distinguished in a definition. In the set theoretic notation (by the axiom of the restricted comprehension) these object constitute a subset. Interesting is the property as formula. If in the formula the collection is used (e.g. for quantification) then such definition is called impredicative. An example of such impredactive definition (based on Goldbach's conjecture) is as follows.
Let n be (the least) even natural number greater than 2 that can not be expressed as the sum of two primes.
The conjecture (there in no such number) still remains unproved. Although the definition is impredicative, it makes sense.

It seems that, in general, an impredicative definition is an abstraction of a type, or of a relation. In the example above the type can be constructed that corresponds to this definition. If in the definition, the requirement (for n to be the least) is relaxed (i.e. removing quantifications), then a relation (an operation of type $N \rightarrow Types$) can be constructed that corresponds to such definition.

An intensional definition, also called a connotative definition, specifies the necessary and sufficient conditions for a thing being a member of a specific set. An extensional definition, also called a denotative definition, of a concept or term specifies its extension. It is a list naming every object that is a member of a specific set. If the collection is finite then intensional and extensional definitions are equivalent.

Definitions are linguistic expressions formed in a (formal) language. A definition is meaningless without a reference to a concrete model (grounding).

8 Continuum and real numbers

A popular view is that a real number is a value that represents a quantity along a continuous line.

The standard axiomatic definition is that real numbers form the unique (up to isomorphism) Archimedean complete totally ordered field. Models of real numbers are based on equivalence classes of Cauchy sequences of rational numbers, Dedekind cuts, and infinite decimal (binary and other) representations (expansions), together with specification of the arithmetic operations, and the linear order. These definitions are equivalent in the classical Mathematics.

Reconstruction of the grounding of Cauchy sequences via integers and rational numbers is intuitively clear and can be done easily. For the Dedekind cuts it is not so clear, however, it also can be done. Dedekind cut is a partition of the set of rational numbers Q into two non-empty sets A and B, such that all elements of A are less than all elements of B, and A has no greatest element. Let Dedekind cut be denoted by (A, B).

Enumeration (with repetitions) of the all elements of the set of rational numbers Q (as an operation of type $N \rightarrow Q$) can be constructed easily and in many ways. Let us fix one operation and denote it as $enum_Q$.

An operation $r : N \rightarrow N$ to be interpreted as a Cauchy sequence must satisfy the following condition.

- for any $n : N$ there exist $m : N$ such that for any $k, l : N$ if $k > m$ and $l > m$, then $n|enum_Q(k) - enum_Q(l)| < 1$.

The above condition can be easily constructed as a relation of type $(N \rightarrow N) \rightarrow Types$, see TO [1]. Of course, it needs some effort to construct multiplication on N and subtraction and linear order on Q. These very relations (each relation corresponds to one Cauchy sequence) are the grounding of the Cauchy sequences in the Universe [1].

To construct a grounding for Dedekind cuts, let us consider partition of Q as operation $p : N \rightarrow N$ such that its output is either 1 (corresponding to A) or 2 (corresponding to B). That is, if $p(n)$ is 1 (resp. 2), then the rational number $enum_Q(n)$ belongs to the set A (resp. to the set B).

For a Dedekind cut (A, B) and the condition

- For any $a \in A$ and any $b \in B$, $\quad a < b$. And there is no $\hat{a} \in A$ such that for all $a \in A$, $\quad a \leq \hat{a}$.

the corresponding formula of type $(N \to N) \to Types$ can be constructed that is the grounding of the abstract notion of *a single* Dedekind cuts in the Universe.

Note that in order to develop Analysis, the mere definitions of Cauchy sequences (Dedekind cuts) are not enough; much more must be done. The Dedekind cuts (as well the set of equivalence classes of Cauchy sequences) must be augmented to form Archimedean complete totally ordered field.

The original grounding of the binary finite representation (expansions) of the real numbers from the unit interval corresponds to the type C^1.

In the context of the above consideration, the grounding of the Continuum proposed by Brouwer and formalized by the introduction of the primitive type C seems to be more straightforward and simpler.

8.1 Continuum according to Brouwer

Recall (see Section 5.1) that C^1 denotes the the type C constructed for the Euclidean patterns of dimension 1, and C_1 consists of two parts denoted by 0 and 1. We are going to show how this type is related to the unit interval $[0; 1]$ of the real numbers considered as binary representations, i.e. finite zero-one sequences.

Since the investigation presented below does not depend on the dimension of the Euclidean patters, let C denote C^n for arbitrary natural number n. If necessary, it will be mentioned that C denotes C^1.

Recall (see Section 3.1) that for c_1 and c_2 of type C, $\quad c_1 \sqsubseteq_C c_2$ means that c_1 is an initial segment of c_2, or is equal to c_2. Operation $r : N \to C$, is called *strictly increasing*, if for any k_1 and k_2, if k_1 is lesser than k_2, then $r(k_1) \sqsubseteq_C r(k_2)$ and $\neg Equal_C(r(k_1); r(k_2))$. Any of such operations corresponds to a Brouwer's law like choice sequence, if for any $k : N$, $r(k)$ is of length k, i.e. belongs to C_k. Note that it means that such operation is a constructible object of type $N \to C$.

Brouwer‘s notion of free choice sequence is not clear. It is an abstraction from types N and C to sets (denoted by $\hat{N}$ and $\hat{C}$), and abstraction from strictly increasing operations of type $N \to C$ to functions from $\hat{N}$ to $\hat{C}$. A free choice sequence is only potentially infinite.

Note that here the adjacency relations is not necessary between objects of the same length. This extension can be easily constructed for any c_1 and c_2 of type C, that is, $Adj_C(c_1; c_2)$ is $Adj^k_C(\bar{c_1}; \bar{c_2})$ such that

- $\bar{c_1}$ and $\bar{c_2}$ are of C_k, and k is the minimum of the length of c_1 and the length c_2,
- and $\bar{c_1}$ is a prefix of c_1, or is equal to c_1,
- and $\bar{c_2}$ is a prefix of c_2, or is equal to c_2.

If C denotes C^1, then a strictly increasing function can be called a pre-real number. Note that the adjacency relation Adj_C determines the equivalence (equality) relations between two pre-real numbers r_1 and r_2, if for all $n : N$, $Adj_C(r_1(n); r_2(n))$. The equivalence classes of the pre-real numbers can be identified with the real numbers from the unit interval. Any equivalence class has exactly two elements (pre-real numbers) for real numbers inside the unit interval, and one element for each border real number, i.e. for zero, and for one.

It is interesting that for such grounding any total function from unit interval into unit interval must be continuous; it is a theorem of Brouwer. To explain this well known *paradox* let us introduce the following definitions.

Operation g of type $C \to C$, is called *monotonic* if for any c_1 and c_2, if $c_1 \sqsubseteq_C c_2$, then $g(c_1) \sqsubseteq_C g(c_2)$.

Operation $g : C \to C$ is called *strictly monotonic* if it is monotonic and for any c_1 and c_2, if $\neg Equal_C(c_1; c_2)$, then $\neg Equal_C(g(c_1); g(c_2))$. Note that for an operation to be interpreted as a function on the unit interval $[0; 1]$, it must be strictly monotonic. However, it is not enough. It must be also continuous.

The *continuity* of an operation $g : C \to C$ is defined as the following condition. For any c_1 and c_2, if $Adj_C(c_1; c_2)$, then $Adj_C(g(c_1); g(c_2))$. This notion of continuity may seem somewhat strange from the classic point of view. However, its intuitive explanation is clear, i.e. if two continua are adjacent (actually they form one single continuum), then after applying a continuous transformation their images are not separated, and also form a single continuum.

To be interpreted as defined on real numbers, an operation must be strictly monotonic. To be function (that is, it cannot have two different values for one argument), it must be also continuous in the sense defined above.

Suppose that the operation in question (say $g : C \to C$) is strictly monotonic, and it is not continuous, i.e. there are two different objects c_1 and c_2, such that $Adj_C(c_1; c_2)$ and $\neg Adj_C(g(c_1); g(c_2))$. Two pre-real numbers (say r_1 and r_2) can be constructed such that $r_1(1)$ is c_1, and $r_2(1)$ is c_2, and for any $n : N$, $Adj_C(r_1(n); r_2(n))$. So that r_1 and r_2 are equivalent, that is, are the same real number. By the strict monotonicity of g, for all $n : N$, $\neg Adj_C(g(r_1(n)); g(r_2(n)))$. It means that the values of the function (determined by operation g) for two equivalent (the same) r_1 and r_2 are different, i.e. not equivalent. Hence, operation g can not be interpreted as function if it is not continuous.

For an operation to be interpreted as a function on the unit interval $[0; 1]$ it must be strictly monotonic, so that it is also continuous. Hence, the function is also continuous in the *classical* sense.

The two groundings of notion of the real numbers (one based on the type C^1, and the second on the binary representations) are related somehow because the binary (finite) representation of rational numbers can be interpreted as objects of type C^1.

By abstraction, the operations of type $C \to C$ may be considered as functions on the set $\hat{C}$.

Every continuous function on the unit interval $[0; 1]$ can be interpolated (represented) by a strictly monotonic (and continuous) function on the set $\hat{C^1}$. It is so because C^1 can be interpreted as a dense subset of the unit interval. The elements of this dense subset are finite zero-one sequences that may be interpreted (like in programming) as rational numbers from the unit interval.

However, the scope of the abstract notion of arbitrary function on $\hat{C^1}$ is much wider than the notion of function on the unit interval.

To be interpreted as function on the unit interval, a function on $\hat{C}^1$ must be strictly monotonic and continuous.

In the case of an abstract complex (see Section 6), it may be interpreted as an operation from C into C taking only values 0 and 1. If it is strictly monotonic, then it determines an abstract (as the limit) manifold. Since it is not continuous, it can not be interpreted as a function on the unit interval. A discontinuity point occurs if the exactly two equivalent pre-reals take different values.

Hence, operations (and functions) on C (on $\hat{C}$) that are strictly monotonic and not continuous are also of interest. Moreover, also some classes of functions (being not necessary strictly monotonic) may also be interesting.

It may suggest the the abstract notion of function on the set of real numbers (as well as the notion of the set of real numbers) does not capture all essential properties of the type C^1.

Note that the type C^1 (and generally, type C) is independent of the notion of number; that is, only the adjacency relation is essential. It seems that the abstraction to the set of real numbers may be restrictive and that Analysis may be developed without points, i.e. without the real numbers.

If the notion of Continuum is identified with the type C^1 (for Euclidean pattern of dimensional 1), then it is different than the notion of the set of real numbers. Operations on Continuum need not be continuous. Usually such operations correspond to approximations on C_k if k is increasing. The limit (actually being an abstraction) may be interpreted as a manifold rather than a function.

The strict monotonicity is the requirement for interpreting such operations as functions on the unit interval of the real numbers. Then, a mapping to be a function must be continuous. This concerns only two sequences that are to be identified as the same real number, however they have different outputs (values). To have discontinuity, arbitrary choice of one of these two outputs is a solution.

Summing up, there is no controversy and no paradox in the Brouwer theorem. It is only misunderstanding related to the interpretation of the grounding of the real numbers and functions on them. The real numbers and the functions are considered in the classic Mathematics as existing objects (having meaning from their existence) that do not need grounding. Actually, they are abstract notions, and without a grounding they are senseless.

The notion of Continuum is independent of the notion of rational numbers. The notion of Continuum is different from the notion of real numbers. The notion of real numbers is an abstract notion grounded in Cauchy sequences of rational numbers. Real numbers can be also grounded in C^1 as the strictly monotonic sequences, i.e. operations of type $N \to C^1$.

9 Another grounding of Analysis

Definitions of the set of real numbers by Cauchy sequences or by Dedekind cuts (that may be seen also as groundings of Continuum) are complex. The definition by decimal (or better binary (dyadic)) representations (expansions) of the real numbers are close to the grounding based on the type C^1.

To developed Analysis, the set of real numbers must be equipped (by additional definitions) with operators for addition, subtraction, multiplications, division, as well as for the linear order and topology. It is difficult and complex.

Analysis can be developed on the basis of C^1; it seems to be quite simple and straightforward. The type C^1 corresponds to the unit interval. Generally, C^n corresponds to the unit cube of dimension n. Extension of C^1 to a type where positive, negative, and arbitrary large real numbers can be interpreted can be done in the following way.

Let us construct the types ΣF^+ and ΣF^-, where F^+ and F^- are of type $N \to Types$ such that for any $n : N, \quad F^+(n)$ and $F^-(n)$ are constant and equal to C^1. Let $\mathbb{C}^1$ denote the disjoin union $\Sigma F^+ + \Sigma F^-$.

Note that any real number has double representation in $\mathbb{C}^1$.

Formally, an object of type $\mathbb{C}^1$ may be denoted in the form $(-, (n, c))$ or $(+, (n, c))$ where minus(-) and plus (+) indicate respectively negative and positive part of the line, $n - 1$ is the integer part whereas c (of type C^1) is the fraction part.

Addition, subtraction, multiplication and simplified division by 2^k applied to objects of $\mathbb{C}^1$ (finite binary representations) giveas a result always objects of type $\mathbb{C}^1$. They can be constructed easily as operations. On the basis of them, the differential and integral calculus can be developed. The relation $Equal_C$ can be easily extended to the operation $Equal_{\mathbb{C}^1}$. Also the adjacency relation Adj can be extended to the type $\mathbb{C}^1$.

For the simplicity of the presentations, let n in $(-,(n,c))$ and $(+,(n,c))$ be represented as binary numbers. Let the operations ADD, SUB of type $(\mathbb{C}^1 \times \mathbb{C}^1) \to \mathbb{C}^1$ denote the arithmetic operations of addition and subtraction. Let DIV_k , and MUL_k of type $\mathbb{C}^1 \to \mathbb{C}^1$ denote respectively the division (resp. multiplication) by 2^k that is done by the shifting of k-positions to the left (resp. to the right) of the separator (comma) between integer part and fraction part of the binary representation of objects of $\mathbb{C}^1$.

9.1 The derivative

Note that for any c, and c_1 and c_2 of $\mathbb{C}^1_k$ such that c_1 (resp. c_2) is left (resp. right) hand adjacent to c the following condition holds:

- $SUB(c_2; c)$ and $SUB(c; c_1)$ considered as binary sequence is the same as the rational number 2^{-k}.

For an operation $f : \mathbb{C}^1 \to \mathbb{C}^1$ its left (right) derivative is constructed as the operation df^- (resp. df^+) of type $\mathbb{C}^1 \to \mathbb{C}^1$, such that

- $df^-(c)$ is $MUL_k(SUB(f(c); f(c_1)))$
- $df^+(c)$ is $MUL_k(SUB(f(c_2); f(c)))$

If operation f is strictly monotonic and continuous, then the left and right derivatives (i.e. df^- and df^+) correspond to the classic notion of the derivative. Note that here the proposed notion is more general and has direct computational meaning.

9.2 The integral

Indefinite integral of operation $g : \mathbb{C} \to \mathbb{C}$ is constructed as the operation $\int g : \mathbb{C} \to \mathbb{C}$ such that for any c of $\mathbb{C}^1_k$, $\int g(c)$ is the same as $DIV_k(g(c))$.

Let $\mathbb{C}^1_k$ consist of the objects of the form $(-, (n, c))$ or $(+, (n, c))$ where c is object of C_k.

For any objects c_0, c_1, c_2, ... , c_{n-1}, c_n, and c_{n+1}, of $\mathbb{C}^1_k$, such that c_0 is left hand adjacent to c_1, and c_1 is left hand adjacent to c_2, ... and c_{n-1} is left hand adjacent to c_n, and c_n is left hand adjacent to c_{n+1}; let the definite integral $\int_{c_n}^{c_1} g$ be the sum of $DIV_k(g(c_1))$, $DIV_k(g(c_2))$, ... , $DIV_k(g(c_n))$.

Note that

$\int_{c_n}^{c_1} df^-$ is equal to $SUB(f(c_n); f(c_0))$, and

$\int_{c_n}^{c_1} df^+$ is equal to $SUB(f(c_{n+1}); f(c_1))$.

Hence, the fundamental theorem of calculus, which states that differentiation is the reverse process to integration, is clearly true and obvious in the proposed framework.

Note that the proposed grounding for Analysis corresponds closely to the Newton-Leibniz approach to infinitesimal calculus introduced in the 17th century.

9.3 Cartesian product

Cartesian product is an abstraction of the product of two types to the product of two sets. Since the set $\mathbb{R}$ is an abstraction, also $\mathbb{R}\times\mathbb{R}$ (denoted also as $\mathbb{R}^2$ in classical Mathematics) does not have direct computational (constructive) meaning.

Note that $\mathbb{R}^2$ (as a topological space) corresponds rather to $\mathbb{C}^2$, than to $\mathbb{C}^1 \times \mathbb{C}^1$. Generally, $\mathbb{R}^n$ corresponds to $\mathbb{C}^n$.

As an instructive example let us construct the type $\mathbb{C}^2$. First, let us construct the types ΣF^{++}, ΣF^{--}, ΣF^{+-}, and ΣF^{-+}, where F^{++}, F^{--}, F^{+-}, and F^{-+} are constant operations of type $(N \times N) \to Types$ with the fixed output equal to the type C^2. Let $\mathbb{C}^2$ denote the disjoin union $\Sigma F^{++} + \Sigma F^{--} + \Sigma F^{+-} + \Sigma F^{-+}$.

An object of type $\mathbb{C}^2$ may be denoted in the form $(--, ((n, m), c))$, or $(++, ((n, m), c))$, or $(-+, ((n, m), c))$, or $(+-, ((n, m), c))$.

The adjacency relation *Adj* (originally for C^2) must be extended (by appropriate construction) to $\mathbb{C}^2$. The extension is done at the borders, so that, ΣF^{++} and ΣF^{--} are not adjacent, as well as ΣF^{+-} and ΣF^{-+}.

Note that each element of $\mathbb{R}^2$ is represented by exactly four objects of $\mathbb{C}^2$.

As a conclusion, let us note that the type $\mathbb{C}^n$ (for $n : N$) is constructed without rational numbers, and it has natural topological structure determined by the adjacent relation. Of course, the objects of $\mathbb{C}^1$ (viewed as finite binary sequences) can be identified with some rational numbers (finite binary extensions), however, it is misleading.

For $\mathbb{C}^2$, (generally, for $\mathbb{C}^n$ where n grater than 1), it is hard, and in fact counterintuitive, to identify its objects with pairs (n-tuples) of some rational numbers. Object of type $\mathbb{C}^n$ cannot be interpreted as a point in the classical sense; it is related to other objects of this type, and it has its internal structure. Points are abstractions and corresponds to the limits of sequences (strictly increasing operations) of objects of type $\mathbb{C}^n$.

Note that $\mathbb{C}^n$ has clear and straightforward computational content. It is important for real calculations, where the scope of the calculations is always finite and determined in advance, before starting the calculations.

10 The crises in Mathematics

This last section expresses some of the author's (rather radical) personal views.

The crisis of Mathematics at the turn of the 19th to the 20th century was concerned with its foundations, and was caused by paradoxes discovered in naive set theory and naive logic.

There were three schools that aimed to give a firm foundations: Logicism (represented by Frege, Russell and Whitehead), Intuitionism (represented mainly by Brouwer), and Formalism (by Hilbert). None of them succeeded. Hermann Weyl [21] wrote in 1946: *... we are less certain than ever about the ultimate foundations of (logic and) mathematics.* The dispute and strong controversy ceased in the next decades. The popular view was: *Don't worry, just do mathematics by proving theorems ...* .

Meanwhile the foundational problem has accrued and becoming more and more severe. In this context, the dispute between Simpson and Voevodsky is interesting (see [17] and [20] for more details).

In 1998, Carlos Simpson published a paper indicating that there might be a mistake in Voevodsky and Kapranov's 1990 result. Voevodsky answered: *Simpson claimed to have constructed a counterexample, but he was not able to show where the mistake was in our paper. ... it was not clear whether we made a mistake somewhere in our paper or he made a mistake somewhere in his counterexample. Mathematical research currently relies on a complex system of mutual trust based on reputations.* In the autumn of 2013 Voevodsky confirmed the error.

As a result, computer proof (checkers) assistants were proposed as trusted tools to resolve such disputes. Currently, some attempts to provide computational foundations for Mathematics emerged, one of them is HoTT [19] based on the theory of types. It is still a formal theory well suited for computer proof (checkers) assistants.

A conclusion may be drown from the above story. Mathematicians don't understand each other! All is based on so called *reputations*. Has the problem of the ultimate foundations for Mathematics returned? If so, then again the following question should be taken seriously. What is Mathematics about? What is type, functions, and relation? What is mathematical reasoning? What is abstraction (abstract objects) in Mathematics?

Otherwise, Mathematics should be vied as a mysterious temple guarded by a few chosen priest of high reputation. If there are serious controversies between the high priests, then a god is needed to judge who is right; it is a software called proof assistant.

Is it a heresy, that Mathematics is simple and beautiful; beautiful because of its ingenious simplicity? According to Errett Bishop [3]: *Mathematics is common sense.*

It seems that Brouwer [18] was right, i.e. logic is a part of Mathematics, language is only for communicating, and mental structures are grounding for mathematical notions.

However, he was right only partially – high level abstractions are ubiquitously used in mathematical reasoning. Hence, the Hilbert's formalism makes sense, however, only if these very abstractions have groundings.

10.1 Conclusion: looking for the lost meaning

The Mathematics as a genuine mental construction created by human minds (developed and cumulated by generations of mathematicians) is beautiful, and ingenious in its simplicity. It can be comprehended not only by mathematicians but also by ordinary people. So called High Math, that is accessible only by a few chosen mathematical genii, is a mystification caused by the formal languages (set theory and category theory) used to describe high abstract mathematical notions, and by the loss of the grounding of these notions.

Formal languages (by definitions and theorems) only roughly describe the essence of the Mathematics, that is, the beauty of the magnificent mental structures. Without grounding (concrete semantics) these languages are mere ... games to play with strings, i.e. they produce more and more formal proofs.

It is instructive that the computational grounding needs careful and detailed constructions to avoid the danger of disconnecting abstractions from their computational grounding, and then to loose this grounding. It is particularly hard for working mathematicians, because these abstractions are natural, simple, and obvious. However, sometimes this simplicity and obviousness are an illusion so that getting to the real grounding needs a strong discipline and introspection, and self control of the reasoning process.

Reconstruction of the grounding of Mathematics is important. Since it was not done so far, it is hard, extremely hard.

Let the final conclusion (related to the grounding of Mathematics) be the famous quotation of David Hilbert: *Wir müssen wissen - wir werden wissen!*

Acknowledgements. The work was supported by the grants:
RobREx - Autonomia dla robotów ratowniczo-eksploracyjnych. Grant NCBR Nr PBS1/A3/8/2012 w ramach Programu Badań Stosowanych w Obszarze Technologie informacyjne, elektronika, automatyka i robotyka, w Ścieżce A.
oraz *IT SOA - Nowe technologie informacyjne dla elektronicznej gospodarki i Społeczeństwa informacyjnego oparte na paradygmacie SOA*; Program Operacyjny Innowacyjna Gospodarka: Działanie 1.3.1.

References

[1] Ambroszkiewicz, S.: Types and operations. ICS PAS Report N. 1030, Warsaw 2014, Preprint `http://arxiv.org/abs/1501.03043`, 2015.

[2] Bell, J.: Divergent conceptions of the continuum in 19th and early 20th century mathematics and philosophy, *Axiomathes*, **1**(15), 2005, 63–84.

[3] Bishop, E.: Quotes, Site on www https://en.wikipedia.org/wiki/Errett_Bishop.

[4] Bishop, E., Bridges, D.: *Constructive Analysis*, Springer Science and Business Media, 1985.

[5] Feferman, S.: Conceptions of the Continuum, *Intellectica*, **51**, 2009, 169–189.

[6] Frege, G.: *Begriffsschrift: eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*, Halle, 1879.

[7] Grzegorczyk, A.: Computable functionals, *Fundamenta Mathematicae*, **42**, 1955, 168–202.

[8] Grzegorczyk, A.: On the definition of computable functionals, *Fundamenta Mathematicae*, **42**, 1955, 232–239.

[9] Grzegorczyk, A.: On the definitions of computable real continuous functions, *Fundamenta Mathematicae*, **44**, 1957, 61–71.

[10] Harper, R.: What is the big deal with? http://existentialtype.wordpress.com/2013/06/22/whats-the-big-deal-with-hott/, 2014.

[11] Hilbert, D.: *Gesammelte Abhandlungen*, vol. 3, Berlin, 1935, Page 159.

[12] Kaczynski, T., Mischaikow, K., Mrozek, M.: Computing homology, *Homology, Homotopy and Applications*, **5**(2), 2003, 233–256.

[13] Kaczynski, T., Mischaikow, K., Mrozek, M.: *Computational Homology*, Springer-Verlag, 2004.

[14] Lacombe, D.: Remarques sur les operateurs recursifs et sur les fonctions recursives d'une variable reelle, *Comptes Rendus de l'Academie des Sciences, Paris*, **241**, 1955, 1250–1252.

[15] Longo, G.: The Mathematical Continuum, From Intuition to Logic, in: *Naturalizing Phenomenology* (Jean Petitot, Franscisco J. Varela, Barnard Pacoud, Jean-Michel Roy, Eds.), Stanford University Press, 1999.

[16] Martin-Löf, P.: An intuitionistic theory of types: predicative part, *Logic Colloqium 1973* (H. E. Rose, J. C. Shepherdson, Eds.).

[17] Roberts, S.: In Mathematics, Mistakes Are not What They Used To Be. Computers can not invent, but they are changing the field anyway, May 15 2015, Online http://nautil.us/issue/24/error/in-mathematics-mistakes-arent-what-they-used-to-be. By Illustration by Richie Pope.

[18] Troelstra, A.: History of constructivism in the 20th century. In Set Theory, Arithmetic, and Foundations of Mathematics. Theorems, Philosophies, *Lecture Notes in Logic*, **36**, 2011, 7–9.

[19] Univalent Foundations Program, T.: *Homotopy Type Theory: Univalent Foundations of Mathematics*, Institute for Advanced Study,http://homotopytypetheory.org/book, 2013.

[20] Voevodsky, V.: The Origins and Motivations of Univalent Foundations, *IAS - The Institute Letter. Summer 2014, Institute for Advanced Study, Princeton, NJ, USA*, 2014, 8–9, Source URL (modified on 11/04/2014): https://www.ias.edu/ias-letter/voevodsky-origins.

[21] Weyl, H.: Mathematics and Logic, *The American Mathematical Monthly*, **53**(1), Jan. 1946, 2–13 (end of the page 13).