Handbook of Bishop Constructive Mathematics

Edited by
Douglas Bridges
Hajime Ishihara
Michael Rathjen
Helmut Schwichtenberg

Contents

Identity, equality, and extensionality in explicit mathematics

Gerhard Jäger
University of Bern

Abstract: A system EC^+ of explicit mathematics is introduced that is conservative over Peano arithmetic PA. This system deals with individuals (called operations) and collections of individuals (called classes). In addition, there is a binary relation $\Re$ acting on individuals and classes, and $\Re(u, U)$ says that the individual u is a name of (or represents) the class U.

We have equality on the level of individuals and equality on the level of classes and an interesting interplay between both. We study some ontological consequences of this interplay, discuss the way how abstract data structures can be represented in EC^+ and consider – as examples – the number systems $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$. This article ends with some first remarks about the representation of the real numbers $\mathbb{R}$.

Keywords: Explicit mathematics, identity, equality, extensionality.

1 Introduction

In the mid seventies Solomon Feferman introduced a new formalism – he called it *explicit mathematics* – motivated by the aim to set up a proper formal framework for *Bishop-style constructive mathematics*, see Bishop (1967). In the three milestone articles Feferman (1975, 1978, 1979) he sketched the main ideas of explicit mathematics, compared his approach with other axiomatic frameworks for constructive mathematics and started to look at explicit mathematics from the broader perspective of proof theory and generalized recursion theory.

This is one of the striking features of explicit mathematics: Although originally designed as a framework for constructive mathematics (and based on intuitionistic logic) it soon became evident that systems of explicit mathematics (now based on intuitionstic or classical logic) play an important role in many other parts of logic (such as, for example, subsystems of second order arithmetic and set theory, reductive proof theory, generalized/abstract recursion theory).

For me personally, explicit mathematics offers a convincing basis for large parts of mathematics. And for me it is not primarily a question of whether one uses intuitionistic or classical logic. Instead, I think that the distinction between concrete and abstract entities is of central importance.

Starting point for us (in this article) is an open ended universe of objects. The exact nature of these objects is intentionally left open; you may think of them as (constructive) operations of any kind, bit strings, computer programs, or whatever. To give these objects a name, we call them *operations*. We only claim that they form a partial combinatory algebra PCA.

Mathematical examples of such universes are Kleene's first and second model, the graph model as well as variants of term models. But one can also imagine situations where application is understood as a more general kind of interaction between objects.

Our universe is open-ended in the sense that new constants can be added whenever needed or useful. The (partial) terms that can be built upon this partial combinatory algebra are our "first-class citizens". Because of the properties of combinatory algebras we have quite some expressive power and a basic computation theory over this base universe.

Classes, on the other hand, are abstractly given collections of operations. But we make sure – by means of a naming relation – that there is a deep connection between the concretely given elements of the PCA at the bottom and the class structure on top of it: every class has to have a name that belongs to the PCA. This naming relation is crucial in our operational approach below.

One of its features is that a sort of higher-type operations become possible. In a nutshell: Level 0 comprises the elements of the PCA; level $n+1$ consists of names of classes whose elements are objects of levels n; and it is obvious how to proceed into the transfinite.

However, from a conceptual point of view it is more important that this sort of naming allows us to look at classes – which are per se extensional objects – also from an *intensional perspective* and as such reconcile set/class-theoretic *Platonism* with a sort of *conceptualism*. The "price" we have to pay is that we have to work with several equality relations. But the "gain" we make is a rich and ontologically interesting structure.

In this article we confine ourselves to a simple system of explicit mathematics with the class of *natural numbers* and *elementary comprehension* as its central class existence principle. Induction on the natural numbers will be restricted to classes so that we stay conservative over Peano arithmetic.

In the next section we introduce the basic axiomatic operational framework. Then we discuss some first ontological consequences. This is followed by some considerations that are related to the identities and equality relations of our formalism. In the following section we treat so-called *abstract data structures* on an abstract level before we turn to specific examples: the number systems $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{Q}$. The last section is dedicated to some remarks about the representation of the structure of the real numbers $\mathbb{R}$ in our formalism.

2 The basic axiomatic operational framework

Before turning to the classes in explicit mathematics we have to look more closely at the general operational framework. We confine ourselves to a relatively weak core operational theory. The basic idea is simple: The universe of discourse is a partial combinatory algebra; its elements are operations and share the following properties:

- Operations may be partial, they may freely be applied to each other, and self-application of operations is permitted.
- As a consequence, the general theory of operations is type-free. Later classes of operations will be added with the purpose to partly structure the universe.
- Operations are intensional objects; extensionality of operations is only assumed or claimed axiomatically in very special situations.

The need to work with possibly undefined objects has also some impact on the logic that we will use; see below.

The system of explicit mathematics with which we will work in the following is formulated in the second order language $\mathbb{L}$ for individuals and classes. It comprises individual variables $a, b, c, f, g, h, u, v, w, x, y, z, \dots$ as well as class variables $U, V, W, X, Y, Z, \dots$ (both possibly with subscripts). $\mathbb{L}$ also includes the individual constants k, s (combinators), $\mathsf{p}, \mathsf{p}_0, \mathsf{p}_1$ (pairing and projections), 0 (zero), s_N (successor), p_N (predecessor), d_N (definition by numerical cases) and additional individual constants, called generators, which will be used for the uniform naming of classes, namely nat (natural numbers), id (identity), co (complement), un (union), dom (domain), and inv (inverse image).

There is one binary function symbol $\cdot$ for (partial) application of individuals to individuals. Further, $\mathbb{L}$ has unary relation symbols $\downarrow$ (defined), N (natural numbers) as well as three binary relation symbols $\in$ (membership), $=$ (equality), and $\Re$ (naming, representation).

The *individual terms* $(r, s, t, r_0, s_0, t_0, \ldots)$ of $\mathbb{L}$ are built up from individual variables and individual constants by means of our function symbol $\cdot$ for application. In the following $\cdot(r, s)$ is usually written as $(r \cdot s)$, (rs) or – if no confusion arises – simply as rs. The convention of association to the left is also adopted so that $r_1 r_2 \ldots r_n$ stands for $(\ldots(r_1 r_2) \ldots r_n)$, and we often also write $s(r_1, \ldots, r_n)$ for $s r_1 \ldots r_n$. $\langle r, s\rangle$ stands for the pair $\mathsf{p}(r, s)$, and general n-tupling is defined by induction on $n \geq 1$ as follows:

$$\langle r_1 \rangle := r_1 \quad \text{and} \quad \langle r_1, \ldots, r_{n+1}\rangle := \langle\langle r_1, \ldots, r_n\rangle, r_{n+1}\rangle.$$

If n is a natural number, we write $\overline{n}$ for the corresponding numeral, i.e., for the closed term given recursively by $\overline{0} :\equiv 0$ and $\overline{n+1} :\equiv \mathsf{s_N}\overline{n}$.

The atomic formulas of $\mathbb{L}$ are the formulas $(r\downarrow)$, $\mathsf{N}(r)$, $(r = s)$, $(r \in U)$, and $\Re(r, U)$. Since we work with a logic of partial terms, it is not guaranteed that all terms have values, and thus $(r\downarrow)$ is used to express that r *is defined* or r *has a value*. Moreover, $\mathsf{N}(r)$ and $(r \in U)$ say that r is a natural number and an element of class U, respectively. Finally, the formula $\Re(r, U)$ is used to express that the individual r *represents* the class U or is a *name* of U.

The *formulas* $(A, B, C, A_0, B_0, C_0, \ldots)$ of $\mathbb{L}$ are generated from these atomic formulas by closing them under the usual propositional connectives and quantification over individuals and classes. We will often omit parentheses if there is no danger of confusion.

An $\mathbb{L}$ formula A is called *stratified* iff the relation symbol $\Re$ does not occur in A; it is called *elementary* iff it is stratified and does not contain bound class variables.

The partial equality relation $\simeq$ is introduced by, following Kleene,

$$(r \simeq s) \;:=\; ((r\downarrow \vee s\downarrow) \;\rightarrow\; r = s),$$

and $(r \neq s)$ is written for $(r\downarrow \wedge s\downarrow \wedge \neg(r = s))$. Hence $(r \neq s)$ is not the logical negation of $(r = s)$.

Since we will be dealing with possibly undefined objects, it is convenient to work with Beeson's *logic of partial terms LPT*, see Beeson (1985), which can be based on classical or intuitionistic logic. It corresponds to the E^+-logic with equality and strictness of Troelsta and van Dalen (1988a), where $E(r)$ is written instead of $r\downarrow$.

Space does not permit to describe *LPT* in detail, but let us mention that the axioms for quantification over individulas have in LPT the form

$$A[r] \wedge r\downarrow \;\rightarrow\; \exists x A[x] \quad \text{and} \quad \forall x A[x] \wedge r\downarrow \;\rightarrow\; A[r].$$

The *definedness axioms* imply that all individual variables and constants are defined and that

$$A[r_1, \ldots, r_n] \;\rightarrow\; r_1\downarrow \wedge \ldots \wedge r_n\downarrow$$

for any atomic A. The *equality axioms* for individuals are as usual, formulated, however, only for variables.[1] Equality for classes is defined (see page 7) and not an axiom.

In the following we work with the classical version of LPT. However, the central arguments go through in intuitionistic LPT as well; if necessary, small adjustments must be made.

Before turning to our systems BON^+ and EC^+ we introduce a few useful shorthand notations:

$$\begin{aligned} r \in \mathsf{N} &:= \mathsf{N}(r), \\ (\exists x \in \mathsf{N})A[x] &:= \exists x(x \in \mathsf{N} \wedge A[x]), \\ (\forall x \in \mathsf{N})A[x] &:= \forall x(x \in \mathsf{N} \rightarrow A[x]), \\ r \in (\mathsf{N}^n \rightarrow \mathsf{N}) &:= (\forall x_1, \ldots, x_n \in \mathsf{N})(r(x_1, \ldots, x_n) \in \mathsf{N}). \end{aligned}$$

Hence $r \in (\mathsf{N}^n \rightarrow \mathsf{N})$ says that the individual r represents an n-ary function from N to N.

All the "usual" systems of explicit mathematics comprise the axioms of a partial combinatory algebra as well as the standard axioms about pairing and projections. In addition, we have some canonical axioms for the natural numbers with successor and predecessor. The following theory BON^+ is the extension of the *basic theory of operations and numbers* BON introduced in Feferman and Jäger (1993) by the schema of induction on the natural numbers for all elementary formulas.

Partial combinatory algebra.

- $\mathsf{k}(x, y) = x$,
- $\mathsf{s}(x, y)\downarrow \wedge \mathsf{s}(x, y, z) \simeq x(z, yz)$.

Pairing and projections.

- $\mathsf{p}_0\langle x, y\rangle = x \;\wedge\; \mathsf{p}_1\langle x, y\rangle = y$.

[1] E.g., $u = u$ is an axiom whereas $r = r$ is false if r is not defined.

Natural numbers.

- $0 \in \mathsf{N} \;\wedge\; \mathsf{s_N} \in (\mathsf{N} \to \mathsf{N})$,
- $\mathsf{s_N}x \neq 0 \;\wedge\; \mathsf{p_N}0 = 0 \;\wedge\; (\forall x \in \mathsf{N})(\mathsf{p_N}(\mathsf{s_N}x) = x)$,
- $A[0] \wedge (\forall x \in \mathsf{N})(A[x] \to A[\mathsf{s_N}x]) \;\to\; (\forall x \in \mathsf{N})A[x]$

for all elementary formulas A. Of course, we could also allow induction for arbitrary $\mathbb{L}$ formulas. However, elementary induction is sufficient for our purposes in this article.

Definition by cases on N.

- $x \in \mathsf{N} \wedge y \in \mathsf{N} \wedge x = y \;\to\; \mathsf{d_N}(a, b, x, y) = a$,
- $x \in \mathsf{N} \wedge y \in \mathsf{N} \wedge x \neq y \;\to\; \mathsf{d_N}(a, b, x, y) = b$,

Since BON^+ comprises the axioms of a partial combinatory algebra, we clearly have λ-abstraction and the usual fixed point theorem. This is mentioned already in Feferman (1975) and proved in detail in, e.g., Beeson (1985) and Troelsta and van Dalen (1988b). Induction on the natural numbers is not needed for these two lemmas.

Lemma 1.1 (λ-abstraction) *For each variable x and term t we can construct a term $\lambda x.t$ whose free variables are those of t, excluding x, such that* BON^+ *proves*

$$\lambda x.t{\downarrow} \;\wedge\; (\lambda x.t)x \simeq t.$$

Lemma 1.2 (Fixed point) *There exists a closed term* fix *such that* BON^+ *proves*

$$\mathsf{fix}(f){\downarrow} \;\wedge\; (g = \mathsf{fix}(f) \;\to\; \forall x(gx \simeq f(g, x))).$$

Following Jäger et al. (2018) we can easily show that BON^+ provides a good framework for dealing with the primitive recursive functions. We write $\mathbf{N}$ for the set of natural numbers. Given a (possibly partial) function $\mathcal{F}$ from $\mathbf{N}^k$ to $\mathbf{N}$ we say that a closed term t *numeralwise represents* $\mathcal{F}$ in BON^+ iff

$$\mathcal{F}(m_1, \ldots, m_k) \simeq n \quad\Longleftrightarrow\quad \mathsf{BON}^+ \;\vdash\; t(\overline{m_1}, \ldots, \overline{m_k}) \simeq \overline{n}$$

for all $m_1, \ldots, m_k, n \in \mathbf{N}$. However, this does not guarantee the expected behavior of t on nonstandard natural numbers. To impose such a condition we have to assume that it is described by formulas, e.g., by equations. For example, let us consider an unary function $\mathcal{G}$ that is defined by primitive recursion from a natural number n_0 and a binary function $\mathcal{F}$ as

$$\mathcal{G}(0) = n_0 \quad\text{and}\quad \mathcal{G}(m + 1) = \mathcal{F}(m, \mathcal{G}(m))$$

for all natural numbers m. Then, if the terms r and s represent the functions $\mathcal{F}$ and $\mathcal{G}$, respectively, we want the conditional equations

$$s0 \simeq \overline{n}_0 \quad \text{and} \quad (\forall x \in \mathsf{N})(s(\mathsf{s_N} x) \simeq r(x, sx)).$$

If the defining formula of a function $\mathcal{F}$ is provable for a term t in BON^+, we say that t *definitionally represents* $\mathcal{F}$ in BON^+. The following is immediate from Troelsta and van Dalen (1988b).

Theorem 1.3 *For any (definition of a) k-ary primitive recursive function $\mathcal{F}$, there exists a closed term $prim_{\mathcal{F}}$ that numeralwise and definitionally represents $\mathcal{F}$ in BON^+ and for which BON^+ proves $prim_{\mathcal{F}} \in (\mathsf{N}^k \to \mathsf{N})$.*

BON^+ is also a reasonable basis for general recursion theory. However, in Jäger et al. (2018) a point is made for introducing a new operation τ_{N}, called *truncation*, to obtain a natural formalization of partial recursive functions and semi-decidability notions in our applicative framework. Adding the non-constructive minimum operator and the Suslin operator allows the move to higher recursion theory; see Feferman and Jäger (1993), Jäger and Strahm (2002), and Jäger and Probst (2011).

3 Adding elementary classes

In this section we turn to classes. As mentioned in the introduction, classes are collections of individuals and will be treated extensionally. Consequently, two classes are considered equal if the have the same elements. Therefore, we define

$$U \subseteq V \;:=\; (\forall x \in U)(x \in V) \quad \text{and} \quad U = V \;:= U \subseteq V \,\wedge\, V \subseteq U,$$

where $(\forall x \in U)A[x]$ is short for $\forall x(x \in U \;\to\; A[x])$.

In our explicit world every class must have a name or – if you prefer this point of view - every class can be addressed via an individual. This is achieved with the help of the relation $\Re$. We claim, in addition, that there are no homonyms and that $\Re$ respects the extensional equality of classes. This gives the next group of axioms.

Explicit representation and equality.

- $\exists x \Re(x, U)$,
- $(\Re(r, U) \,\wedge\, \Re(r, V)) \;\to\; U = V$,
- $(U = V \,\wedge\, \Re(r, U)) \;\to\; \Re(r, V)$.

We say that an individual r is a *name* iff there exists a class X which is named by r; individual r "belongs" to individual s iff s is the name of a class that contains r. Also, r is equal in this new sense to s iff r and s name the same class. From now on we use the following notations:

$$\begin{aligned}
\Re(r) &:= \exists X \Re(r, X), \\
r \dot{\in} s &:= \exists X(\Re(s, X) \wedge r \in X), \\
r \doteq s &:= \exists X(\Re(r, X) \wedge \Re(s, X)), \\
(\exists x \dot{\in} r)A[x] &:= \exists x(x \dot{\in} r \wedge A[x]), \\
(\forall x \dot{\in} r)A[x] &:= \forall x(x \dot{\in} r \rightarrow A[x]) \\
r \dot{\subseteq} s &:= (\forall x \dot{\in} r)(x \dot{\in} s).
\end{aligned}$$

So, if $\Re(r)$ and $\Re(s)$, we clearly have

$$r \doteq s \leftrightarrow (r \dot{\subseteq} s \wedge s \dot{\subseteq} r).$$

If the vector $\vec{r}$ consists of the individual terms $r_1, \ldots, r_n$ and the vector $\vec{U}$ of the class variables $U_1, \ldots, U_n$, then

$$\Re(\vec{r}, \vec{U}) := \Re(r_1, U_1) \wedge \ldots \wedge \Re(r_n, U_n).$$

Instead of $\Re(r)$ we often write $r \in \Re$. However, this somewhat sloppy notation must not give the impression that $\Re$ is a class. Classes (more precisely: elementary definable classes) are generated by the following axioms.

Basic class existence axioms. In the following we provide a finite axiomatization of uniform elementary comprehension.

Natural numbers

- $\mathsf{nat} \in \Re$,
- $\forall x(x \dot{\in} \mathsf{nat} \leftrightarrow \mathsf{N}(x))$.

Identity

- $\mathsf{id} \in \Re$,
- $\forall x(x \dot{\in} \mathsf{id} \leftrightarrow \exists y(x = \langle y, y \rangle))$.

Complements

- $a \in \Re \leftrightarrow \mathsf{co}(a) \in \Re$,
- $a \in \Re \rightarrow \forall x(x \dot{\in} \mathsf{co}(a) \leftrightarrow \neg(x \dot{\in} a))$.

Unions

- $a, b \in \Re \leftrightarrow \mathsf{un}(a,b) \in \Re$,
- $a, b \in \Re \rightarrow \forall x(x \dot{\in} \mathsf{un}(a,b) \leftrightarrow x \dot{\in} a \vee x \dot{\in} b)$.

Domains

- $a \in \Re \leftrightarrow \mathsf{dom}(a) \in \Re$,
- $a \in \Re \rightarrow \forall x(x \dot{\in} \mathsf{dom}(a) \leftrightarrow \exists y(\langle x, y\rangle \dot{\in} a))$.

Inverse images

- $a \in \Re \leftrightarrow \mathsf{inv}(a,f) \in \Re$,
- $a \in \Re \rightarrow \forall x(x \dot{\in} \mathsf{inv}(a,f) \leftrightarrow fx \dot{\in} a)$.

These are the generators and axioms if we work in classical logic. In the intuitionistic case more such generators and axioms are needed, for example, a generator and axioms for intersections.

Our theory EC^+ for elementary classes is the theory BON^+ extended by the axioms for explicit representation and equality plus the basic class existence axioms. The plus signifies that induction on N for elementary formulas is available. Accordingly, EC is EC^+ without the schema of induction on N for elementary formulas.

That the naming "elementary classes" is justified becomes evident in view of the following theorem. It is taken from Feferman and Jäger (1996) and states that (informally written) the class $\{x : A[x, \vec{v}, \vec{W}]\}$ can be uniformly named by a term $t_A(\vec{v}, \vec{w})$, provided that $\vec{w}$ names the classes $\vec{W}$.

Theorem 1.4 (Elementary comprehension) *For every elementary formula $A[u, \vec{v}, \vec{W}]$ with at most the indicated free variables there exists a closed term t_A such that one can prove in* EC*:*

(1) $\Re(\vec{w}, \vec{W}) \rightarrow t_A(\vec{v}, \vec{w}) \in \Re$,
(2) $\Re(\vec{w}, \vec{W}) \rightarrow \forall x(x \dot{\in} t_A(\vec{v}, \vec{w}) \leftrightarrow A[x, \vec{v}, \vec{W}])$.

In view of this theorem EC^+ is equivalent to EC plus the axiom of induction on N,

$$\forall X(0 \in X \wedge (\forall y \in \mathsf{N})(y \in X \rightarrow \mathsf{s_N} y \in X) \rightarrow (\forall y \in \mathsf{N})(y \in X)). \quad (\mathsf{C\text{-}I_N})$$

It also follows from Feferman and Jäger (1996) that EC^+ is a conservative extension of Peano arithmetic. Until the end of this article we will concentrate on EC and EC^+, discuss some basic ontological properties of this framework and show that (and how) abstract data structures present themselves there.

4 About some ontological aspects of EC and EC^+

There are two sorts of ontological properties of EC: (i) those referring to the first order part of EC^+, and (ii) those that have to do with classes. We begin with the first group and consider three additional principles.

- *Operational extensionality* (*Op-Ext*): $\forall x(fx \simeq gx) \to f = g$.
- *Totality* (T): $\forall x, y(xy\downarrow)$.
- *All operations are numbers* (N): $\forall x(x \in \mathsf{N})$.

Each of these principles makes strong ontological claims which will not be considered natural in most cases. Nevertheless, it is interesting to see how they relate to each other. The following assertions are spread over the literature and can be found, for example, in Jäger (2017).

Theorem 1.5 *None of the principles* (*Op-Ext*), (T), *or* (N) *is provable in* EC^+. *In addition, we have:*

(1) $\mathsf{EC}^+ + (\textit{Op-Ext}) + (T)$ *is consistent.*
(2) $\mathsf{EC} + (\textit{Op-Ext}) + (N)$ *is inconsistent.*
(3) $\mathsf{EC} + (T) + (N)$ *is inconsistent.*

There are further first order principles that would deserve attention[2] but we cannot discuss them here and refer to the literature; see, e.g., Jäger (2017) and Jäger et al. (2018). Instead, we turn to ontological properties that have to do with the class structure of EC.

Let us begin these consideration with the observation that EC plus full comprehension is inconsistent. It is proved by mimicking the usual Russell argument.

Theorem 1.6 EC *plus full comprehension is inconsistent.*

Proof Given full comprehension, there exists a class U such that

$$\forall x(x \in U \leftrightarrow \exists X(\Re(x, X) \wedge x \notin X)).$$

According to our axioms about explicit representation and equality, U has a name, say u. These axioms also tell us that all classes named u are extensionally equal, therefore

$$\exists X(\Re(u, X) \wedge u \notin X) \leftrightarrow u \notin U.$$

Together with the previous equivalence, this yields that $u \in U$ iff $u \notin U$. A contradiction. □

[2] Full definition by cases and truncation are such principles.

Of course, this result is not surprising. It can be shown, however, that EC plus comprehension for stratified formulas is consistent.

Now we turn to one of the most central properties of the naming relation. It says that the names of a class never form a class. The following is a generalization of Theorem 3 of Jäger (1979).

Theorem 1.7 $\mathsf{EC} \vdash \forall X \neg \exists Y \forall z(z \in Y \leftrightarrow \Re(z, X))$.

Proof Working informally in EC, let U be an arbitrary class and u one of its names. We consider the elementary formula

$$A[x, y, V, W] \;:=\; (x \notin V \;\wedge\; y \notin W) \;\vee\; (x \in V \;\wedge\; y \in W).$$

In view of Theorem 1.4 there exists a closed term t_A such that

$$\Re(v, V) \wedge \Re(w, W) \;\rightarrow\; t_A(y, v, w) \in \Re, \tag{1}$$

$$\Re(v, V) \wedge \Re(w, W) \;\rightarrow\; \forall x(x \mathrel{\dot{\in}} t_A(y, v, w) \;\leftrightarrow\; A[x, y, V, W]). \tag{2}$$

We set

$$s \;:=\; \lambda w.t_A(w, u, w)$$

and conclude from (1) and (2) that, for any w and W,

$$\Re(w, W) \quad \rightarrow \quad sw \;\doteq\; \begin{cases} u & \text{if } w \in W, \\ \mathsf{co}(u) & \text{if } w \notin W. \end{cases} \tag{3}$$

Now assume that the names of U form a class Z_0, i.e.

$$\forall x(x \in Z_0 \;\leftrightarrow\; \Re(x, U)),$$

and that Z_1 is introduced by elementary comprehension such that

$$\forall x(x \in Z_1 \;\leftrightarrow\; sx \notin Z_0).$$

Therefore,

$$\forall x(x \in Z_1 \;\leftrightarrow\; \neg\Re(sx, U)). \tag{4}$$

In addition, let z_1 be a name of Z_1. Then (3) yields

$$sz_1 \;\doteq\; \begin{cases} u & \text{if } z_1 \in Z_1, \\ \mathsf{co}(u) & \text{if } z_1 \notin Z_1. \end{cases} \tag{5}$$

From (4) and (5) we obtain that $z_1 \in Z_1$ iff $z_1 \notin Z_1$. This is a contradiction, implying that the names of U cannot form a class. □

Let U and V be classes with the names u and v, respectively. Then it is clear that $u = v$ implies $U = V$ and thus $u \doteq v$. However, in general it is possible that a class U has two different names, i.e. we may have $u \neq v$ but $u \doteq v$. By adding class extensionality, we would rule this possibility out, but, as it turns out, class extensionality is not compatible with EC.

Theorem 1.8 EC *plus* class extensionality

$$(\forall x, y \in \Re)(x \doteq y \rightarrow x = y) \qquad (\textit{Cl-Ext})$$

is inconsistent.

Proof Pick, e.g., the class of natural numbers. From (*Cl-Ext*) we could derive that all names of this class are identical to nat and thus form a class (by elementary comprehension), contradicting our previous theorem. □

Although the names of a class never form a class, it is consistent with EC^+ to claim that there exists the class of all names. This can be seen by extending the model construction for EC that is presented in detail in Feferman (1979).

Theorem 1.9 *The assertion* $\exists X \forall x(x \in X \leftrightarrow x \in \Re)$ *is consistent with* EC^+*, but not provable in* EC^+*.*

With some additional effort even a strengthening of this result is possible: We can consistently assume in EC that all objects are names.

Power classes provide some problems in explicit mathematics. The naive approach is to claim that that for every class X there exists a class Y such that Y contains exactly the names of <u>all</u> subclasses of X. We call it the *strong power class axiom*:

$$\forall X \exists Y \forall z(z \in Y \leftrightarrow \exists Z(\Re(z, Z) \wedge Z \subseteq X)). \qquad (\mathsf{SP})$$

The *weak power class axiom* asks for less: It only claims that for each class X there exists a class Y such that each element of Y names a subclass of X and for any subclass of X <u>at least one</u> of its names belongs to Y,

$$\forall X \exists Y((\forall z \in Y)(\exists Z \subseteq X)(\Re(z, Z)) \wedge (\forall Z \subseteq X)(\exists z \in Y)\Re(z, Z)). \qquad (\mathsf{WP})$$

Both, the strong and the weak power class axiom, are problematic. By Theorem 1.7 we know that in EC the names of the empty class cannot form a class, and thus the strong power class of the empty class cannot exist.

Corollary 1.10 (SP) *is inconsistent with* EC*.*

The weak power class axiom is less problematic in the sense that it is consistent with EC. However, one may rightly argue whether it is in the spirit of explicit mathematics.

Unlike in ordinary mathematics, explicit mathematics does not permit the identification of power classes with the collection of its characteristic operations. Given a class U neither the strong nor the weak power class of U make much sense. On the other hand,

$$\{0,1\}^U \ := \ \{f : (\forall x \in U)(fx = 0 \lor fx = 1)\}$$

is a class by elementary comprehension. In most cases it is the adequate "substitute" for the power class of U. This is also the approach taken in Section 7 below.

5 Abstract data structures

As we have seen, there are already two equality relations on our basic universe of individuals:

- $a = b$ means that a and b are identical in the sense of the underlying partial combinatory algebra. It is the strongest form of equality in explicit mathematics. Whenever $a = b$ then a will be equal to b also in the sense of any equality relation to be introduced later.
- $a \doteq b$ means that a and b name the same class. So $\doteq$ is inherited from the extensional equality of classes: If a is a name of class U and b is a name of class V then $a \doteq b$ iff U and V are extensionally equal as classes.

These two equality relations are "global" equality relations on the universe. However, in many situations "local" equalities are required. This is exactly what Bishop and Bridges do when they introduce their notion of set; see Bishop and Bridges (1985), pp. 15–16:

The totality of all mathematical objects constructed in accordance with certain requirements is called a set. *The requirements of the construction, which vary with the set under consideration, determine the set. Thus the integers form a set, the rational numbers form a set, and … the collection of all sequences of integers is a set.*

Each set will be endowed with a binary relation of equality. *This relation is a matter of convention, except that it must be an* equivalence relation.

This approach can be implemented in EC in a very natural way. Given $a \in \Re$, we write $a \times a$ for a name of the class of all pairs $\langle x, y \rangle$ with $x, y \,\dot{\in}\, a$. Such a name is provided by elementary comprehension.

Definition 1.11 A *set* is defined to be a pair $a = \langle a_1, a_2 \rangle$ such that $a_1, a_2 \in \Re$ and $a_2 \dot{\subseteq} a_1 \times a_1$ is an equivalence relation on a_1. In this case we call a_1 the *universe* and a_2 the *equality* of a.

In the following we let small boldface Latin letters range over sets. Then $|\mathbf{a}|$ stands for the universe of $\mathbf{a}$ and $=_{\mathbf{a}}$ for its equality; i.e.

$$\mathbf{a} \;=\; \langle |\mathbf{a}|, =_{\mathbf{a}} \rangle.$$

Clearly, we write $x =_{\mathbf{a}} y$ instead of $\langle x, y \rangle \dot{\in} =_{\mathbf{a}}$ for all $x, y \dot{\in} |\mathbf{a}|$. Also, if $\vec{u}$ and $\vec{v}$ are the lists $u_0, \ldots, u_n$ and $v_1, \ldots, v_n$ of elements of $|\mathbf{a}|$, then $\vec{u} =_{\mathbf{a}} \vec{v}$ is short for

$$u_1 =_{\mathbf{a}} v_1 \;\wedge\; \ldots \;\wedge u_n =_{\mathbf{a}} v_n.$$

Keep in mind that sets are represented via their names. In general, we can have that $|\mathbf{a}| \doteq |\mathbf{b}|$ and, for any $x, y \dot{\in} |\mathbf{a}|$,

$$x =_{\mathbf{a}} y \;\leftrightarrow\; x =_{\mathbf{b}} y$$

although $\mathbf{a}$ and $\mathbf{b}$ are different as names.

Therefore, a higher equality on the level of sets can be introduced: For sets $\mathbf{a}$ and $\mathbf{b}$ we set

$$\mathbf{a} =_{set} \mathbf{b} \quad := \quad \begin{cases} |\mathbf{a}| \doteq |\mathbf{b}| \;\wedge \\ (\forall x, y \dot{\in} |\mathbf{a}|)(x =_{\mathbf{a}} y \;\leftrightarrow\; x =_{\mathbf{b}} y). \end{cases}$$

However, for the following we do not need this equality.

There are two operations on sets that are particularly important: *Cartesian products* and *Cartesian powers*. In view of Theorem 1.4 it is easy to see that there is a closed term $\mathsf{carprod}$ such that $\mathsf{carprod}(\mathbf{a}, \mathbf{b})$ is, for all sets $\mathbf{a}$ and $\mathbf{b}$, the pair $\langle |\mathbf{a}| \times |\mathbf{b}|, =_{\mathbf{a} \times \mathbf{b}} \rangle$ where $=_{\mathbf{a} \times \mathbf{b}}$ stands for the binary relation on $|\mathbf{a}| \times |\mathbf{b}|$ given by, for all $x_1, x_2 \dot{\in} |\mathbf{a}|$ and $y_1, y_2 \dot{\in} |\mathbf{b}|$,

$$\langle x_1, y_1 \rangle =_{\mathbf{a} \times \mathbf{b}} \langle x_2, y_2 \rangle \;\leftrightarrow\; x_1 =_{\mathbf{a}} x_2 \;\wedge\; y_1 =_{\mathbf{b}} y_2.$$

In the following we simply write $\mathbf{a} \times \mathbf{b}$ for $\mathsf{carprod}(\mathbf{a}, \mathbf{b})$. $\mathbf{a}^2$ is then defined to be the set $\mathbf{a} \times \mathbf{a}$, and $\mathbf{a}^{n+1} = \mathbf{a}^n \times \mathbf{a}$.

The introduction of the Cartesian power requires a bit more care. First we make use of Theorem 1.4 to find a closed term efun[3] such that $\mathsf{efun}(\mathbf{a}, \mathbf{b})$ names the class

$$\{f : (\forall x \dot{\in} |\mathbf{a}|)(f(x) \in |\mathbf{b}|) \;\wedge\; (\forall x, y \dot{\in} |\mathbf{a}|)(x =_{\mathbf{a}} y \;\rightarrow\; f(x) =_{\mathbf{b}} f(y))\}$$

[3] It stands for "extenional functions".

of all functions from $|\mathbf{a}|$ to $|\mathbf{b}|$ that respect the equalities of $\mathbf{a}$ and $\mathbf{b}$. The standard equivalence relation on $\mathsf{efun}(\mathbf{a}, \mathbf{b})$ is then given by

$$f =_{\mathbf{a}\to\mathbf{b}} g \;\leftrightarrow\; (\forall x \,\dot{\in}\, |\mathbf{a}|)(f(x) =_{\mathbf{b}} g(x))$$

and thus elementary. We write $\mathbf{b}^{\mathbf{a}}$ for the pair $\langle \mathsf{efun}(\mathbf{a}, \mathbf{b}), =_{\mathbf{a}\to\mathbf{b}} \rangle$ and observe that $\mathbf{b}^{\mathbf{a}}$ is a set.

In a next step we look at functions and relations acting on a given set. f is called an *n-extensional function* on set $\mathbf{a}$ if $f \in (|\mathbf{a}|^n \to |\mathbf{a}|)$ and for all strings $\vec{x}$ and $\vec{y}$ of length n the following condition is satisfied:

$$\vec{x} \,\dot{\in}\, |\mathbf{a}| \wedge \vec{x} =_{\mathbf{a}} \vec{y} \;\to\; f(\vec{x}) =_{\mathbf{a}} f(\vec{y}).$$

Accordingly, r is called an *n-extensional relation* on set $\mathbf{a}$ if $r \,\dot{\subseteq}\, |\mathbf{a}|^n$ and for all strings $\vec{x}$ and $\vec{y}$ of length n the following condition is satisfied:

$$\vec{x} =_{\mathbf{a}} \vec{y} \wedge \langle \vec{x} \rangle \,\dot{\in}\, r \;\to\; \langle \vec{y} \rangle \,\dot{\in}\, r.$$

Thus the n-extensional functions on a set $\mathbf{a}$ are exactly those n-ary total functions on $\mathbf{a}$ which respect its equality; analogously for the n-extensional relations on $\mathbf{a}$. Such functions and relations together with the underlying set form what we call an abstract data structure.

Definition 1.12 An *abstract data structure* consists of a set $\mathbf{a}$ and finitely many functions and relations of various arities that are extensional on $\mathbf{a}$ and thus is a tuple of the form $\langle \mathbf{a}, f_1, \ldots, f_m, r_1, \ldots, r_n \rangle$.

Rather than studying abstract data structures on a general level we now turn to a series of specific such structure.

6 The number systems $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{Q}$ as abstract data structures

In the case of the natural numbers the situation is simple: The natural numbers themselves are directly given in explicit mathematics, and the corresponding equality relation is the identity relation of our language $\mathbb{L}$. Consequently, the *set of natural numbers* in our sense is given as the pair

$$\mathbf{nat} \;:=\; \langle \mathsf{nat}, \mathsf{id} \rangle.$$

As far as functions and relations on $\mathbf{nat}$ are concerned, we can work with the usual primitive recursive machinery, which is available in EC^+ according to Theorem 1.3. Thus the algebraic properties of the natural numbers are summarized in the abstract data structure

$$\mathbb{N} \ := \ \langle \mathbf{nat}, 0, 1, +_{\mathsf{N}}, \cdot_{\mathsf{N}}, <_{\mathsf{N}} \rangle$$

that describes an ordered commutative semi-ring without zero divisors in which 0 is the least element and 1 its successor. It should be evident that $(\mathsf{C}\text{-}\mathsf{I}_{\mathsf{N}})$ is sufficient for proving the basic arithmetic properties of $\mathbb{N}$.

Any integer is coded as a pair $\langle x, y \rangle$ of natural numbers, intended to represent the difference $x - y$. Again we proceed by elementary comprehension and fix a constant term int such that

$$\mathsf{int} \ \doteq \ \{\langle x, y \rangle : \mathsf{N}(x) \wedge \mathsf{N}(y)\}.$$

and a constant term $=_{\mathsf{Z}}$ for integer equality, satisfying

$$\langle x_1, y_1 \rangle =_{\mathsf{Z}} \langle x_2, y_2 \rangle \ \leftrightarrow \ x_1 +_{\mathsf{N}} y_2 \ = \ x_2 +_{\mathsf{N}} y_1$$

for all $x_1, x_2, y_1, y_2 \in \mathsf{N}$. It is obvious that

$$\mathbf{int} \ := \ \langle \mathsf{int}, =_{\mathsf{Z}} \rangle$$

is a set. The zero element 0_{Z} and unit element 1_{Z} for **int** are defined by $\langle 0, 0 \rangle$ and $\langle 1, 0 \rangle$, respectively. The addition $+_{\mathsf{Z}}$, subtraction $-_{\mathsf{Z}}$, and multiplication $\cdot_{\mathsf{Z}}$ are 2-extensional functions on **int** defined such that, for $x_1, x_2, y_1, y_2 \dot{\in} \mathsf{nat}$,

$$\langle x_1, y_1 \rangle +_{\mathsf{Z}} \langle x_2, y_2 \rangle \ = \ \langle x_1 +_{\mathsf{N}} x_2, y_1 +_{\mathsf{N}} y_2 \rangle,$$

$$\langle x_1, y_1 \rangle -_{\mathsf{Z}} \langle x_2, y_2 \rangle \ = \ \langle x_1 +_{\mathsf{N}} y_2, y_1 +_{\mathsf{N}} x_2 \rangle,$$

$$\langle x_1, y_1 \rangle \cdot_{\mathsf{Z}} \langle x_2, y_2 \rangle \ = \ \langle (x_1 \cdot_{\mathsf{N}} x_2) +_{\mathsf{N}} (y_1 \cdot_{\mathsf{N}} y_2), (y_1 \cdot_{\mathsf{N}} x_2) +_{\mathsf{N}} (x_1 \cdot_{\mathsf{N}} y_2) \rangle.$$

Finally, there is an elementary definable and 2-extensional less-relation on **int** with

$$\langle x_1, y_1 \rangle <_{\mathsf{Z}} \langle x_2, y_2 \rangle \ \leftrightarrow \ x_1 +_{\mathsf{N}} y_2 \ <_{\mathsf{N}} \ x_2 +_{\mathsf{N}} y_1$$

again for all $x_1, x_2, y_1, y_2 \in \mathsf{N}$. Then it can be seen easily that the abstract data structure

$$\mathbb{Z} \ := \ \langle \mathbf{int}, 0_{\mathsf{Z}}, 1_{\mathsf{Z}}, +_{\mathsf{Z}}, \cdot_{\mathsf{Z}}, <_{\mathsf{Z}} \rangle$$

is an ordered integral domain. We have an injection of $\mathbb{N}$ into $\mathbb{Z}$ by mapping any $x \dot{\in} \mathsf{nat}$ onto $\langle x, 0 \rangle$.

The rational numbers are introduced by following a similar path. Any rational number is coded as a pair $\langle x, y \rangle$ with $x, y \dot{\in} \mathsf{int}$ and y different from 0_{Z}. As before, we work in EC, fix a constant rat such that

$$\mathsf{rat} \ \doteq \ \{\langle x, y \rangle : x, y \dot{\in} \mathsf{int} \ \wedge \ y \neq_{\mathsf{Z}} 0_{\mathsf{Z}}\}$$

and introduce a further constant $=_\mathsf{Q}$ for the equality relation on int, satisfying

$$\langle x_1, y_1\rangle =_\mathsf{Q} \langle x_2, y_2\rangle \;\leftrightarrow\; x_1 \cdot_\mathsf{Z} y_2 =_\mathsf{Z} y_1 \cdot_\mathsf{Z} x_2$$

for all $x_1, x_2, y_1, y_2 \dot{\in} \mathsf{int}$ where y_1 and y_2 are different from 0_Z. Observe that

$$\mathbf{rat} \;:=\; \langle \mathsf{rat}, =_\mathsf{Q}\rangle$$

is a set with the corresponding zero element $0_\mathsf{Q} := \langle 0_\mathsf{Z}, 1_\mathsf{Z}\rangle$ and unit element $1_\mathsf{Q} := \langle 1_\mathsf{Z}, 1_\mathsf{Z}\rangle$.

As before it is now straightforward to introduce 2-extensional functions for addition, subtraction and multiplication and a 2-extensional less relation on **rat** that satisfy that

$$\begin{aligned}
\langle x_1, y_1\rangle +_\mathsf{Q} \langle x_2, y_2\rangle &= \langle (x_1 \cdot_\mathsf{Z} y_2) +_\mathsf{Z} (x_2 \cdot_\mathsf{Z} y_1), y_1 \cdot_Z y_2\rangle, \\
\langle x_1, y_1\rangle -_\mathsf{Q} \langle x_2, y_2\rangle &= \langle (x_1 \cdot_\mathsf{Z} y_2) -_\mathsf{Z} (x_2 \cdot_\mathsf{Z} y_1), y_1 \cdot_Z y_2\rangle, \\
\langle x_1, y_1\rangle \cdot_\mathsf{Q} \langle x_2, y_2\rangle &= \langle x_1 \cdot_\mathsf{Z} x_2, y_1 \cdot_Z y_2\rangle, \\
\langle x_1, y_1\rangle <_\mathsf{Q} \langle x_2, y_2\rangle &\leftrightarrow \begin{cases} (x_1 \cdot_\mathsf{Z} y_2 <_\mathsf{Z} x_2 \cdot_\mathsf{Z} y_1 \wedge 0_\mathsf{Z} <_\mathsf{Z} y_1 \cdot_\mathsf{Z} y_2) \vee \\ (x_2 \cdot_\mathsf{Z} y_1 <_\mathsf{Z} x_1 \cdot_\mathsf{Z} y_2 \wedge y_1 \cdot_\mathsf{Z} y_2 <_\mathsf{Z} 0_\mathsf{Z}), \end{cases}
\end{aligned}$$

where $\langle x_1, y_1\rangle$ and $\langle x_2, y_2\rangle$ range over the elements of rat. Now it can be verified in EC^+ that

$$\mathbb{Q} \;:=\; \langle \mathbf{rat}, 0_\mathsf{Q}, 1_\mathsf{Q}, +_\mathsf{Q}, \cdot_\mathsf{Q}, <_\mathsf{Q}\rangle$$

is an ordered field. $\mathbb{Z}$ is embedded into $\mathbb{Q}$ by sending x to $\langle x, 1_\mathsf{Z}\rangle$. In particular, for every natural number x greater than 0 its quotient $1/x$ is easily represented in $\mathbb{Q}$ by setting

$$x^{-1} \;:=\; \langle 1_\mathsf{Z}, \langle x, 0\rangle\rangle.$$

If x is 0, then x^{-1} does of course not belong to rat.

Before turning to the representation of the real numbers in the next section, we have to address the treatment of *absolute values* of integers and rationals. For the integers this is achieved by the 1-extensional function $|.|_\mathsf{Z}$ on **int** with

$$|\langle x, y\rangle|_\mathsf{Z} \;=\; \begin{cases} \langle x, y\rangle & \text{if } y <_\mathsf{N} x, \\ \langle y, x\rangle & \text{if } y \not<_\mathsf{N} x \end{cases}$$

for all $x, y \dot{\in} \mathsf{nat}$. This function is easily defined by means of definition by integer cases and the closed term $\mathsf{cut_{off}}$ for the binary primitive-recursive cut-off subtraction; simply set

$$|u|_{\mathsf{Z}} \;:=\; \mathsf{d_N}(\langle \mathsf{p_1}u, \mathsf{p_0}u\rangle, u, \mathsf{cut_{off}}(\mathsf{p_0}u, \mathsf{p_1}u), 0).$$

The absolute value of a rational number $\langle x, y\rangle$ is given by

$$|\langle x, y\rangle|_{\mathsf{Q}} \;:=\; \langle |x|_{\mathsf{Z}}, |y|_{\mathsf{Z}}\rangle.$$

It should be clear the the functions $|.|_{\mathsf{Z}}$ and $|.|_{\mathsf{Q}}$ are extensional on **int** and **rat**, respectively.

In the following, when we work with functions extensional on **int** or **rat** and when there is no ambiguity as determined by the context, we drop the subscripts 'Z' and 'Q'.

7 Representing the real numbers

So far the build up of the number systems $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{Q}$ has been completely canonical – perhaps with the small peculiarity that we make use of specific equality relations rather than moving to equivalence classes. - and there has not been much choice how to proceed. When it comes to the real numbers the situation is different and one can take different paths, for example:

- Dedekind cuts,
- Cauchy sequences,
- nestings of intervals with rational bounds,
- completion of the topological group of rational numbers.

As it turns out the approach to the reals in Bishop and Bridges (1985) can be reproduced very well in explicit mathematics. This has been worked out up to a to certain extent in Feferman (2012).[4] In order to illustrate the conceptual power of the explicit framework, we recall a few central definitions.

Let nat^+ be a name of the class of all positive natural numbers,

$$\mathsf{nat}^+ \;\doteq\; \{x : \mathsf{N}(x) \wedge x \neq 0\}.$$

A *sequence* of elements of a class named a is an operation $x \in (\mathsf{nat}^+ \to a)$. In this case we often write x_n for $x(n)$ and $(x_n : n \,\dot{\in}\, \mathsf{nat}^+)$ for x.

[4] Unfortunately, this research note is not available via Feferman's homepage. If you are interested in it, please contact me.

Definition 1.13

(1) A sequence of rational numbers is called *regular* iff for all $m, n \,\dot{\in}\, \mathsf{nat}^+$,

$$|x_m - x_n| \leq m^{-1} + n^{-1}.$$

(2) By a *real number* is meant a regular sequence of rational numbers;

$$\mathsf{real} \;\doteq\; \{x : x \,\dot{\in}\, (\mathsf{nat}^+ \to \mathsf{rat}) \wedge x \text{ is regular}\}$$

(3) For the equality relation on real we introduce a further constant $=_{\mathsf{R}}$ such that

$$x =_{\mathsf{R}} y \;\leftrightarrow\; (\forall n \in \mathsf{nat}^+)(|x_n - y_n| \leq 2 \cdot n^{-1})$$

for all $x, y \,\dot{\in}\, \mathsf{real}$.

(4) Finally, we introduce the pair **real** and observe that it is a set;

$$\mathbf{real} \;:=\; \langle \mathsf{real}, =_{\mathsf{R}} \rangle.$$

We end this section with introducing a series of functions that are extensional on the set **real**. These definitions are taken from Feferman (2012) and are based on Bishop and Bridges (1985).

Associated with each real number $x = (x_n : n \,\dot{\in}\, \mathsf{nat}^+)$ is a *canoniocal bound* $\mathbf{k}_x$ such that

$$|x_n| \;<\; \mathbf{k}_x$$

for all $n \,\dot{\in}\, \mathsf{nat}^+$; $\mathbf{k}_x$ may simply be taken to be the least integer greater than $|x_1| + 2$. Working informally, one may now proceed with defining, for all $x, y \,\dot{\in}\, \mathsf{real}$:

$$\begin{aligned}
x + y &:= (x_{2n} + y_{2n} : n \,\dot{\in}\, \mathsf{nat}^+), \\
x \cdot y &:= (x_{2kn} \cdot y_{2kn} : n \,\dot{\in}\, \mathsf{nat}^+) \text{ for } k = \max(\mathbf{k}_x, \mathbf{k}_y), \\
\max(x, y) &:= (\max(x_n, y_n) : n \,\dot{\in}\, \mathsf{nat}^+), \\
-x &:= (-x_n : n \,\dot{\in}\, \mathsf{nat}^+), \\
|x| &:= \max(x, -x).
\end{aligned}$$

It has to be checked that the sequences $x+y$, $x \cdot y$, $max(x, y)$, and $-x$ belong to real and that these functions are extensional on **real**. It is a routine matter to verify that this can be done in EC^+.

It should be clear how to embed the rational numbers into the reals: Send a $q \,\dot{\in}\, \mathsf{rat}$ simply to the regular sequence $q^* = (q, q, \ldots)$. Then operations like addition and multiplication are easily seen to be preserved under this embedding.

One aspect that is distinctive to constructivity and goes beyond explicit representations is the definition of positivity of real numbers. The real number x is defined to be *positive* iff for some positive integer k, $k^{-1} < x_k$. On the other hand, a real number x is called *non-negative* iff $-n^{-1} < x_n$ for all $n \mathrel{\dot{\in}} \mathsf{nat}^+$.

Based on that the ordering on the reals is given by

$$y < x \ := \ (x - y) \text{ is positive,}$$

$$y \leq x \ := \ (x - y) \text{ is non-negative.}$$

Calling, in addition, a real number x *non-zero* iff its absolute value is positive, completes the picture.

We are now in the realm of "ordinary" constructive analysis and set

$$\mathbb{R} \ := \ \langle \mathbf{real}, 0^*, 1^*, +, \cdot, \max, \leq, <, \ldots \rangle$$

for the reals and the usual operations on them. For more details about the ordering of real numbers, the question of non-zeroness, the least upper bound principle, and continuous functions we refer to Feferman (2012).

One should consider the conceptual considerations above and the technical work Feferman (2012) as a proof of principal that a reasonable part of elementary analysis can be developed in a weak fragment of explicit mathematics such as EC^+ – a conservative extension of Peano arithmetic. There are also many interesting extensions of EC^+ by, for example, the join axiom, inductive generation, higher functionals and/or strong induction principles that provide suitable frameworks for stronger parts of mathematics such as complex analysis or measure theory.

References

Beeson, M. J. 1985. *Foundations of Constructive Mathematics*. Ergebnisse der Mathematik und ihrer Grenzgebiete, vol. 3/6, Springer.

Bishop, E. 1967. *Foundations of Constructive Analysis*. McGraw-Hill Series in Higher Mathematics, McGraw-Hill.

Bishop, E. and Bridges, D. S. 1985. *Constructive Analysis*. Grundlehren der mathematischen Wissenschaften, vol. 279, Springer.

Feferman, S. 1975. A language and axioms for explicit mathematics. Pages 87–139 of: *Algebra and Logic*. (Crossley, J. N.,ed.), Lecture Notes in Mathematics, vol. 450, Springer.

Feferman, S. 1978. Recursion theory and set theory: a marriage of convenience. Pages 55–98 of: *Generalized Recursion Theory II, Oslo 1977* (Fenstad, J. E., R. O. Gandy R. O., and Sacks, G. E., eds.), Studies in Logic and the Foundations of Mathematics, vol. 94, Elsevier.

Feferman, S. 1979. Constructive theories of functions and classes. Pages 159–224 of: *Logic Colloquium '78* (Boffa, M.,van Dalen, D., and McAloon, K., eds.), Studies in Logic and the Foundations of Mathematics, vol. 97, Elsevier.

Feferman, S. 2012. Constructive analysis in a *BON* system. *Technical note*, Stanford University.

Feferman, S. and Jäger, G. 1993. Systems of explicit mathematics with non-constructive μ-operator. Part I. *Annals of Pure and Applied Logic*, **65**, no. 3, 243–263.

Feferman, S. and Jäger, G. 1996. Systems of explicit mathematics with non-constructive μ-operator and join. Part II *Annals of Pure and Applied Logic*, **79**, no. 1, 37–52.

Jäger, G. 1997. Power types in explicit mathematics?. *The Journal of Symbolic Logic*, **62**, no. 4,1142–1146.

Jäger, G. 2017. The operational penumbra: some ontological aspects. Pages 253–283 of: *Feferman on Foundations: Logic, Mathematics, Philosophy* (Jäger G. and and Sieg, W., eds.), Outstanding Contributions to Logic, vol. 13, Springer.

Jäger, G. and Probst, D. 2011. The Suslin operator in applicative theories: Its proof-theoretic analysis via ordinal theories *Annals of Pure and Applied Logic*, **162**, no. 8, 647–660.

Jäger, G, Rosebrock, T., and Sato, K. 2018. Truncation and semi-decidability notions in applicative theories. *The Journal of Symbolic Logic*, **83**, no. 3, 967–990.

Jäger, G. and Strahm, T. 2002. The proof-theoretic strength of the Suslin operator in applicative theories. Pages 270–292 of: *Reflections on the Foundations of Mathematics: Essays in Honor of Solomon Feferman* (Sieg W., Sommer R, and Talcott, C., eds.), Lecture Notes in Logic, vol. 15, Association for Symbolic Logic.

Troelstra A. S., and van Dalen, D. 1988a *Constructivism in mathematics, I*, Studies in Logic and the Foundations of Mathematics, vol. 121, Elsevier.

Troelstra A. S., and van Dalen, D. 1988b *Constructivism in mathematics, II*, Studies in Logic and the Foundations of Mathematics, vol. 123, Elsevier.