University of Amsterdam
Theory of Computer Science

Division by Zero and Abstract Data Types

J.A. Bergstra

Report TCS1404 February 2014

J.A. Bergstra

section Theory of Computer Science
Faculty of Science
University of Amsterdam

Science Park 904
1098 XH   Amsterdam
the Netherlands

tel. +31 20 525.7591
e-mail: J.A.Bergstra@uva.nl

Theory of Computer Science Electronic Report Series

Division by Zero and Abstract Data Types

Jan A. Bergstra

Section Theory of Computer Science, Informatics Institute, University of Amsterdam
email: j.a.bergstra@uva.nl,janaldertb@gmail.com

Abstract

A survey is given of issues regarding division by zero, in particular in connection with the presentation of elementary mathematics. The viewpoint is defended that accepting $\frac{1}{0}$ as a valid expression is harmless, and that setting its value equal to 0 is useful.

Keywords and phrases: Division by zero, irrelevant syntax, cancellation meadow, partial cancellation meadow, short-circuit logic.

1 Introduction

Division by zero (DbZ) is a remarkable issue and, although its relevance may be denied without hesitation by professional mathematicians, as a topic it seems to survive in the periphery of applied and educational mathematics, and to some extent in computer science. In this paper I will provide a survey of the topic and I will introduce what I call the standard view on DbZ, as well as an alternative to that viewpoint which I will prefer. The alternative is based the theory of abstract data types.

Taking abstract datatype theory as a point of departure, task at hand is to design abstract data types that are closest to the mathematical practice of calculation with values in fields. I will need to explain in detail why in my opinion the known mathematical practice cannot be taken for the required solution, and what it is that the data type perspective adds in this particular issue.

To begin with I will list some questions pertaining to division by zero, which emerge from observing mathematical practice:

1. Is there anything that mathematicians know about the "issue of division by zero" that laymen are likely not to know and which explains that non-mathematicians are more inclined to worry about the matter?

   For instance mathematicians may claim relevant knowledge of the concept of a partial function, or of "how division is properly defined", or of "the concept of an abbreviation". Or they may have some classification of "undefined potential entities" at hand of which division by zero produces merely an example not in need of any additional elaboration because of its simplicity.

2. How can it be that the general knowledge that in a field $x \cdot 0 = 1$ has no solution creates a setting in which $\frac{1}{0}$ should not be written (instead of that knowledge merely being an argument for its non-existence), whereas the observation that above every natural number there is a larger one does not stand in the way of speaking of "the largest natural number" by stating that "there is no largest natural number"?

   Stated in other words: is "the largest natural number" equally undefined as "one divided by zero" or is it for some reason less problematic, as may be witnessed by the fact that the assertion "there is no largest natural number" is considered acceptable as an expression of knowledge able for being professionally communicated in exactly that form? What may be the grounds for such differences when present at all?

3. Can one imagine a viable version of mathematics (mathematical practice) where the question "what is division by zero" is thought of as a question just as much entitled to an answer as the question "what is exponentation in the complex numbers".

4. What makes the difference between the "professionally acceptable" assertion that "one cannot divide 1 by 0" and its slightly more "formalistic" versions "$\frac{1}{0}$ does not exist" or "$\frac{1}{0}$ is undefined" (both of which are less likely to occur in a professional mathematical text)?

5. What (if any) is the difference in status between the non-existence of $\frac{1}{0}$ and the non-existence of $\sum_{n=1}^{\infty} \frac{1}{n}$?

   It seems that writing that "$\frac{1}{0}$ is undefined" is less acceptable (plausible, professional, informative) than writing that "$\sum_{n=1}^{\infty} \frac{1}{n}$ is undefined". The knowledge that $\frac{1}{0}$ does not exist constitutes a part of the basic mathematical communication skills to such an extent that even writing that "$\frac{1}{0}$ does not exist" makes an unprofessional impression, while writing that "$\sum_{n=1}^{\infty} \frac{1}{n}$ does not exist" expresses elementary knowledge able of being expressed in exactly that form.

6. Assuming that data type theory has been used to redesign the story of fields and numbers systems featuring the issues of division by zero: what lesson can data type theory "learn" from the conventional mathematical style which seems not to be in need of the precision which is though to be the crux and the key advantage of data types.

I will not provide answers to these questions because such answers depend too much on one's view on the matter. The questions are merely meant as an incentive for thinking about the topic at hand.

Structure of the paper. Section 2 provides a display of my perception of the dominant position regarding division by zero which I will formulate explicitly as "the standard view on DbZ". That view constitutes a combination of what I will call "the standard view on irrelevant syntax" and the viewpoint that $\frac{1}{0}$ in particular represents an instance of irrelevant syntax. In Section 3 three examples are discussed of syntax in which DbZ plays a role. In Section 4 the signature of meadows is mentioned as a way to deal with DbZ in a systematic fashion. Various advantages and disadvantages of that move are mentioned and the virtues of meadows are outlined. Finally Section 5 contains some concluding remarks.

2 A conventional position on division by zero (CPDbZ)

I will now attempt to collect aspects of a conventional viewpoint on DbZ. CPDbZ is based on the normally implicit but nevertheless crucially important principle that:

> "all mathematical writing takes place in the context of and with the objective of achieving a most efficient and effective communication of facts."

I will call this assumption the *communicative goal assumption*. Consequences of the communicative goal principle are these:

1. Following the communicative goal principle writing an expression just because one's so-called syntax allows for it is considered bad style and lacks legitimacy.
2. By writing an expression t one takes full responsibility for the claim that considering t is worth of the reader's time. In view of the communicative goal assumption a very plausible reply to the question "what is $\frac{1}{0}$" runs as follows: "you started talking about this issue, not me, so you should know best, therefore don't ask me'.
3. If the question had been to explain what will be the result of "dividing an airplane by a bicycle" asking for further explanation and clarification from the person asking that question, before making any attempt to answer the question, would have been almost unavoidable, and entirely reasonable. Indeed in the game of exchanging mathematical information both sides actively take responsibilities by making assertions, and such responsibilities cannot be implicitly burdened on one's opponent in a debate.

These important conventions are communicated at once and almost irreversibly through the single issue of division by zero. In practical terms CPDbZ may be identified with the following viewpoints:

1. There is a notion of a problematic occurrence of DbZ in a text. Such occurrences must be avoided by authors and must be spotted and criticized by readers. Here is an example of a problematic occurrence of DbZ:

   "define $f\colon \mathbb{Q} \to \mathbb{Q}$ by:
   $f(x) = \dfrac{x+1}{x^2-4}$ if $x \notin \{-2, 2\}$, and $f(x) = \dfrac{x^3-5}{x+2}$ otherwise."
2. An author following CPDbZ believes that he or she knows how to write mathematical texts without problematic occurrences of DbZ.

   For instance in $x - x \neq 0 \to \frac{b}{x-x} = c$, with x ranging over the rational numbers, DbZ is avoided by means of a (counterfactual) relevant implication and for that reason what might be DbZ is considered unproblematic. In the very similar case $x \neq x \rightarrow \frac{b}{x-x} = c$ one may argue that DbZ is avoided as a consequence of the use of a material implication.

3. When faced with an expression t containing a problematic occurrence of DbZ, say $t \equiv \frac{1}{0}$ a CPDbZ compliant reader notices that there is no mathematical entity known to the reader which he or she would prefer to denote by writing t.

4. Finally P concludes that lacking any incentive to make use of expressions like t, the expression t may be classified as a piece of irrelevant syntax and for that reason its meaning need not be studied and need not be regarded to be part of mathematics.

5. Questions about expressions in (what P considers) irrelevant syntax are considered (by P) to be irrelevant questions as well.

6. CPDbZ suggests not to perceive a compelling need for formal syntax in one's preferred philosophy of mathematics.[1]

7. When asked for a CPDbZ compliant formalization in a logical style, an author following CPDbZ may wish to introduce a third truth value, say $\perp$ for "meaningless", and to interpret expressions involving (problematic occurrences of) DbZ as undefined and equations and relations involving undefined "expressions" as $\perp$.

   By extending the logical connectives as follows: $\texttt{true} \vee \perp = \perp \vee \texttt{true} = \texttt{true}, \texttt{true} \wedge \perp = \perp \wedge \texttt{true} = \perp, \texttt{false} \vee \perp = \perp \vee \texttt{false} = \perp, \texttt{false} \wedge \perp = \perp \wedge \texttt{false} = \texttt{false}$, and $\neg\perp = \perp$, a setting is obtained where for instance for all $x \in \mathbb{Q}$, $(x \neq 0 \rightarrow \frac{1}{x} \neq 0) \Leftrightarrow (\neg(x \neq 0) \vee \frac{1}{x} \neq 0) \Leftrightarrow \texttt{true}$.

8. However, when contemplating the following assertion Φ with x and y ranging over $\mathbb{Q}$,

   "for all x either $\dfrac{1-x}{1-x} = 1$ or $\dfrac{1+x}{1+x} = 1$"[2]

   and its candidate formalization $\forall x \cdot (\frac{1-x}{1-x} = 1 \vee \frac{1+x}{1+x} = 1)$, someone subscribing to CPDbZ may conclude that when instantiating the latter assertion with $x = 1$ an "unsafe" reading results.

   This observation then calls for a less permissive formalization that makes use of sequential logic primitives instead: $\forall x \cdot (\dfrac{1-x}{1-x} = 1 \mathbin{{}^{\circ}\!\vee} \dfrac{1+x}{1+x} = 1)$.[3] The logical formula $\dfrac{1-x}{1-x} = 1 \mathbin{{}^{\circ}\!\vee} \dfrac{1+x}{1+x} = 1$ evaluates to $\mathbf{a}$ at $x = 1$. Under this second formalization convention the assertion Φ is rejected because it implicitly features a DbZ fault.

[1] Just as much as point sizes of symbols don't enter the foundations of mathematics, formal syntax with named constants, functions, relations, and sorts, is primarily understood as a tool mounted when things have to be "industrialized", for instance in preparation of the design of software support, or when preparing exercises and examples for a logic course.

[2] This assertion is meaningful to the extent that it holds in fields of characteristic not equal to 2.

[3] Implication is expressed via negation and disjunction as usual. Sequential logic primitives originate from [15]. The asymmetric notation for sequential logic primitives where a circle indicates which argument must be evaluated first was introduced in [3]. Sequential logic primitives have many different interpretations, (for instance see [6]). Under the name short-circuit logic the meta-theory of sequential primitives has been studied in detail in [9].

> The CPDbZ approach to avoiding problematic occurrences of DbZ may the be summarized as as follows: "write texts and logical formulae in such a way that a plausible reading via sequential primitives with certainty won't produce **a** as the top-level evaluation result, and expect other authors to proceed accordingly".

I will refer to the communication goal assumption combined with the the first 7 items above as the conventional position on DbZ (CPDbZ). Item 8 is not included in CPDbZ because many principals may be disinclined to contemplate such examples (being considered irrelevant idiom rather than irrelevant syntax) or may not see the necessity to introduce asymmetric connectives in order to formalize such examples.

The opinions of mathematicians are less uniform on this matter in practice, and I cannot exclude that some mathematicians may consider what I have called CPDbZ to represent a caricature of their own views about the issue.

2.1 A conventional position on irrelevant syntax (CPIS)

There are many more sources of irrelevant syntax than DbZ. As an example consider a setting where one uses mathematical symbols for Dollars as well as for Euros. In some ad hoc syntax Dollars and Euros might be multiplied. The resulting expression may be classified as irrelevant syntax by an observer P.

A refusal of P to contemplate the multiplication of Dollars and Euros might be based on a generalization of the CPDbZ to the case of arbitrary irrelevant syntax. That generalization I will call the conventional position on irrelevant syntax (CPIS). CPIS is very simple: seemingly serious questions about (expressions in) irrelevant syntax are perceived as irrelevant questions.

Now the CPDbZ might be decomposed into (i) CPIS and, (ii) the point of view that $\frac{1}{0}$ constitutes irrelevant syntax. A combination of both positions implies CPDbZ.

I consider the conventional position on irrelevant syntax to be an unavoidable position, and the existence of irrelevant syntax to be a fact of life that cannot be denied, even if one knows how to avoid its use as an author by adhering to good practices.

In other words: CPIS is valid when applied to irrelevant syntax.

2.2 Criticizing the classification of $\frac{1}{0}$ as irrelevant syntax

My criticism on these positions (CPDbZ and CPIS) is not their intrinsic logical weakness, because there is no such weakness. Mathematicians, just as the members of any other professional community, need a way for not responding to questions that they consider entirely immaterial. Returning to the example mentioned in 2.1, I do not advocate that a mathematician should be willing to multiply Dollars with Euros, or even to contemplate that kind of "operation", just because someone presents notations for all three components of such a potential "entity" and proposes terms combining these parts into a structure said to represent a multiplication of two of these components.

However, a criticism that can be raised against the CPDbZ is that this particular instantiation of CPIS is unconvincing. In other words I criticize the view that $\frac{1}{0}$ should be labeled as irrelevant syntax. My criticism of labeling $\frac{1}{0}$ as irrelevant syntax is meant in an economic sense, that is in the sense that I subscribe to the viewpoint that a limited part of mathematical practice would become more effective, or productive, if that viewpoint were to be reversed.

2.3 A signature oriented position on division by zero (MSPDbZ)

A signature oriented position on DbZ introduces the notion of a signature as an essential element, thereby producing full legitimacy for semantic questions about fractional expressions. A signature lists names for sorts and for constants and names for operations coupled with so-called arities, that is sequences of (names of) sorts from which successive arguments are taken, as well as the name of a sort where output is delivered.

The suggestion that I will advocate is to take $\frac{1}{0}$ seriously as a piece of mathematical notation, that is not to qualify it as irrelevant syntax. The signature of meadows provides names for a sort, for the two constants zero and one, for addition and for additive inverse, and finally for multiplication and for multiplicative inverse. MSPDbZ is the meadow signature position on DbZ. Adopting MSPDbZ implies the denial of CPIS in the case of (at least some) problematic DbZ occurrences. Reasons for doing so are diverse and will be covered in Section 3.

The problem that CPDbZ produces from the perspective of data types is twofold: (1) the missing concept of a signature (or its abstract version as used in mathematical logic: the similarity type), and (2) granted that 0,1, $+$, $\cdot$, and - are considered function symbols, the absence of a function symbol for inverse (or division) in spite of its abundant use in mathematical practice.

In other words a contribution that data type theory may add to the topic of division by zero is this:

1. Fields like $\mathbb{Q}$ are best seen as abstract data types.
2. All data types (including rings and fields) have a signature. The signature must be explanatory for the usage of the data type.
3. Once a signature has been fixed a range of different interpretations of sorts, constants, and functions may still be open. This range includes the interpretation of function names as partial functions.
4. Partiality of an operator is no ground for its deletion from a signature, and,
5. Partiality of an operator is no ground for classifying informal expressions where the operator is seemingly applied to arguments outside its domain as irrelevant syntax.

2.4 Abstract data type design based positions on DbZ

MSPDbZ needs to be refined with semantic considerations. That amounts to the design of one or more abstract data types which can serve as carriers for working in various number systems.

Although rings and fields are classical algebraic structures research focusing on data types that capture their key aspects is still recent. As an initial paper on the matter I mention [11] which provides an initial algebra abstract data type specification of the meadow of rational numbers.[4]

At this stage there is no ground for claiming that meadows constitute an optimal data type for providing the meaning of the names of the meadow signature. Partial meadows or error-totalized meadows, perhaps equipped with three-valued or four-valued logics making use of sequential connectives, might turn out to provide more natural carriers for accruing out the practices of elementary mathematics.

3 Reasonable occurrences of DbZ

Here are three simple examples indicative of complications of the view that all forms of division by zero lead to irrelevant syntax. In each of these cases I hold that admitting that division by zero produces a valid expression simplifies the picture, though at the price of moving towards either a non-classical logic (three or more truth values), or a logic of partial functions, or some form of relevant implication, or an interpretation of mathematical natural language where `false` $\to \rho$ is considered valid (as a material implication) even in the case that ρ consists of (or contains) irrelevant syntax.[5]

3.1 Quadratic equations

Consider the following assertion: if $a \neq 0$ then the equation $a \cdot x^2 + b \cdot x + c = 0$ has the following solutions x_1 and x_2:

$$x_1 = \frac{-b - \sqrt{b^2 - 4\,a \cdot c}}{2\,a}, x_2 = \frac{-b + \sqrt{b^2 - 4\,a \cdot c}}{2\,a}.$$

From this piece of elementary mathematics we infer:
$\phi \equiv$ "If $a \neq 0$ then $a \cdot (\frac{-b - \sqrt{b^2 - 4\,a \cdot c}}{2\,a})^2 + b \cdot (\frac{-b - \sqrt{b^2 - 4\,a \cdot c}}{2\,a}) + c = 0$." The assertion ϕ is a household fact. Its formalization in first order logic reads:
$\forall a\,(a \neq 0 \to a \cdot (\frac{-b - \sqrt{b^2 - 4\,a \cdot c}}{2\,a})^2 + b \cdot (\frac{-b - \sqrt{b^2 - 4\,a \cdot c}}{2\,a}) + c = 0)$.

We assume that a mathematician intends to check ϕ. Checking the validity of ϕ requires checking the validity of all of its substitution instances, in particular the instance obtained by substituting 0 for a: $0 \neq 0 \to 0 \cdot (\frac{-b - \sqrt{b^2 - 4 \cdot 0 \cdot c}}{2 \cdot 0})^2 + b \cdot (\frac{-b - \sqrt{b^2 - 4 \cdot 0 \cdot c}}{2 \cdot 0}) + c = 0$. When working in a two-valued logic checking the validity requires that it can be ensured that evaluating the equation $0 \cdot (\frac{-b - \sqrt{b^2 - 4 \cdot 0 \cdot c}}{2 \cdot 0})^2 + b \cdot (\frac{-b - \sqrt{b^2 - 4 \cdot 0 \cdot c}}{2 \cdot 0}) + c = 0$ produces either `true` or `false`. For the result of the evaluation of ϕ it does not matter which case

[4] In the meadow of rational numbers inverse is made total by setting $0^{-1} = 0$.

[5] An example of ρ is found from item 7 in Section 2 by substituting $x = 0$ in $x \neq 0 \to \frac{1}{x} = 5$. Short-circuit logic ([15, 3, 6, 9, 7]) is an example of a logic with this quality.

applies, but in fact both cases are equally implausible under the assumption that division by 0 leads to an undefined result. One finds that for a very plausible assertion ϕ either its formalization in first order logic deviates from the most plausible form, or leads to an assertion that cannot be validated by means of an interpretation in a two-valued logic.

If the given formalization in first order logic must be rejected, the question how a compositional meaning assignment must be found for ordinary texts stands out as being in need of further explanation. If three of more truth-values must be applied the question arises which logic must be used. A reasonable way to formalize ϕ is through short-circuit logic (see [6, 7, 9]). That formalizes ϕ as $\forall a\,(a = 0 \mathbin{\overset{\circ}{\vee}} a \cdot (\frac{-b - \sqrt{b^2 - 4\,a \cdot c}}{2\,a})^2 + b \cdot (\frac{-b - \sqrt{b^2 - 4\,a \cdot c}}{2\,a}) + c = 0)$.

As mentioned above (Section 2 item 8) in short-circuit logic one may maintain that $\texttt{true} \mathbin{\overset{\circ}{\vee}} \rho$ evaluates to `true` even if ρ constitutes irrelevant syntax.

3.2 Conditional probability

If $P(b) > 0$ then the conditional probability of a with respect to b is defined as follows:

$$P(a \mid b) = \frac{P(a \wedge b)}{P(b)}.$$

The viewpoint that this definition is a speech act which is performed only if $P(b) > 0$ creates a dynamic setting where different paths of reading lead to different sets of definitions having been "made". Though conceivable in principle I consider this way of reading conditional definitions unconvincing. Instead I hold that:

- The conditional probability $P(a \mid b)$ is undefined if $P(b) = 0$.
- One cannot hold in general for all a, b that $P(a \mid b) = P(a \mid b)$ unless one subscribes to the truth of $\frac{1}{0} = \frac{1}{0}$ as well.
- The standard view on DbZ indicates that $P(a \mid b)$ is a partial function of events a and b. For that reason a systematic logic of conditional probabilities needs to be a logic of partial functions.

3.3 Bayes' theorem

Bayes' theorem is often phrased as follows: $P(x \mid y) = \frac{P(y \mid x) \cdot P(x)}{P(y)}$. The condition that $P(y) \neq 0$ if often not mentioned, though when $P(y) = 0$ the rule seems to make no sense under the standard view on DbZ. Now one might think that

(ψ) "If $P(y) = 0$ then $P(x \mid y) = \frac{P(y \mid x) \cdot P(x)}{P(y)}$"

is a valid statement. However, if $P(x) = 0$ (and $P(y) = 0$) then $P(x \mid y) = 0$ and thus, using ψ, we find that $0 = P(x \mid y) = \frac{P(y \mid x) \cdot P(x)}{P(y)} = \frac{P(y \mid x) \cdot 0}{P(y)} = P(y \mid x) \cdot \frac{0}{P(y)} = P(y \mid x) \cdot 0$. This fact is problematic because $P(y \mid x)$ is not defined. We must conclude that (under the standard view on DbZ) both $P(x) \neq 0$ and $P(y) \neq 0$ are needed as preconditions for Bayes' theorem.[6]

It seems to be the case that many presentations and uses of Bayes' theorem do without a bookkeeping of conditions that guarantee that division by zero is avoided. It seems to be the case that all renderings of Bayes' theorem failing to mention these non-zeroness conditions are invalid under the standard view on DbZ.

4 Working out the meadow-syntax position on DbZ

There are many alternatives to the viewpoint that the notation $\frac{1}{0}$ constitutes irrelevant syntax. We assume that we are working in the field $\mathbb{Q}$ of rational numbers. Here is a survey of some options:

1. Introduction of a division operator and/or an inverse operator to the family of operator names that are allowed for expression formation.[7] Now $\frac{1}{0}$ becomes an expression, (that is a non-irrelevant or valid expression). Having granted some additional (formerly irrelevant) expressions official status one may still leave the way in which (some of) those expressions is dealt with unspecified. Then it is acknowledged that some principled questions are left unanswered, but that acknowledgement is admittedly not justified by means of discarding such questions as ill-posed.[8]

   I will label this alternative *open admission of DbZ expressions*.[9] In [4] the signature of fields extended with an inverse operator is termed the signature of meadows with inversive notation, whereas the signature of fields extended with a division operator is termed the signature of meadows with divisive notation. Thus open admission of DbZ expressions amounts to adopting meadow syntax in either of these versions without further constraining the fate of undefined values such as $\frac{1}{0}$.

2. There are many options for the subsequent refinement of the position of open admission of DbZ expressions, that is for giving a meaning to the signature of meadows. For instance one may work with short-circuit logic in combination with known algebraic transformations and only hold true what is provable. This option depends heavily on

[6]Both conditions are mentioned for instance on `http://www.proofwiki.org/wiki/Bayes'_Theorem`. However, the number of sites and documents on Bayes' theorem (or Bayes' rule according to quite a number of authors who claim, usually without giving an argument, that the fact is not a theorem) that fail to mention such conditions is remarkable.

[7]It seems fair to say that mathematicians have a limited appreciation of the concept of syntax. The strategy of open admission of DbZ syntax (denial of DbZ expression being irrelevant) includes the introduction of syntax (names for constants, operations and sorts) as a key element which seems to be entirely foreign to the mathematical mind in spite of significant steps made in mathematical logic, all depending on turning syntax into an additional first class citizen of the conceptual universe.

[8]A significant advantage of this approach is that wrong answers to such questions need not be sold as valid by professionals who must defend that $\frac{1}{0}$ presents irrelevant syntax without having thought through the details of that position.

[9]Open refers to the open ended character of the admission of expressions previously seen as irrelevant. By means of subsequent refinement the admission may be come less "open". Open admission of DbZ expressions includes the permission to use expressions $\frac{p}{q}$ without mentioning guarantees that $q \neq 0$.

the choice of a particular proof theory. For that reason its complexity is problematic and other options as mentioned below are more plausible.

3. Working with partial functions and with a syntax for partial functions. Then $\frac{1}{0}$ is a valid expression for which $\frac{1}{0} = \frac{1}{0}$ may not evaluate to `true`. A logic of partial functions may be in need of a non-classical logic with at least third truth-value.

4. Introduction of an error value (say $\mathbf{a}_{\mathbb{Q}}$ when the field is $\mathbb{Q}$) external to the domain of the field at hand, and extension of its domain with that error value, withs subsequent extension of the graph of all operations. In this case a non-classical logic is plausible as well.

5. Totalization of division by means of an arbitrary choice of value, for instance by setting $\frac{1}{0} = 17$.

6. Totalization of division by a less arbitrary choice of value for the expression $\frac{1}{0}$ in particular by setting $\frac{1}{0} = 0$. These design decisions underly the theory of meadows for which I refer to [11, 13, 4, 1]. Earlier work in the same direction can be found in [14] and [16].

4.1 Relative advantages and disadvantages of alternatives to the standard view on DbZ

In [4] it has been outlined that designing logics of partial functions or logics of error algebra extensions for the theory of fields, or even for a single specific field, is not a simple matter and that the details of such developments must not be underestimated.

It is concluded that the solution of meadows (with DbZ producing 0), viewed as a refinement of the open admission of DbZ expressions and compared with other such refinements, seems to be the simplest option in terms of the development cost and the expected educational transmission cost of the meta-theory that results.

I conclude that open admission of DbZ expressions is amply justified by the abundance of its use in practice. I conclude in addition that the choice of a refinement which enables to answer the various questions that an open admission of DbZ expression generates is unavoidable for systematic projects on exposition and education.

The choice of meadows is a matter of finding a cheapest refinement of open admission of DbZ expressions (that is working with the syntax of meadows or an extension of that syntax) which is complete in the sense that it answers all or most “DbZ questions”. The argument is an economic one rather than a principled one. And the economic merits of the argument still need confirmation. Finding that confirmation may take very long, and it may never occur if additional benefits of other refinement strategies provide a “competitive advantage” of those alternatives (against the application of meadows) on the long run.

4.2 Meadows: an axiomatic approach to the refinement of open admission of DbZ expressions?

The theory of meadows begins with [11] where the term "meadow" was introduced. When that paper was published we were unaware of [14] and [16] which (at this moment) I consider to be "true" original sources of the equations for meadows. Here are some results that have been obtained with meadows:

1. The equational theory of meadows (called pseudo-fields in [14, 16]) is decidable ([14]).
2. Development of an initial algebra specification of the meadow of rationals $\mathbb{Q}_0$ viewed as an abstract data type ([11], and subsequently improved in [4]).
3. The initial algebra of the equational theory of meadows is characterized in [13].
4. Basis theorems for meadows and extension theories ([1], generalizing results in [14] and [16]).
5. A finite equational basis is found in [2] for the equations true in the meadow of reals and also for the equations that hold in the meadow of complex numbers with complex conjugation.
6. Axioms and related definitions for probability functions that work without non-zeroness assumptions ([8]).
7. In [12] a method is described by which mistakes that might be introduced (seen from a conventional point of view) by computing in a meadow are detected and corrected.
8. The relevant division convention of [4] combines the application of meadows for providing a rigorous semantic model of the language of elementary mathematics with a conventional style of working with conditions in such a way that only relevant use (that is not dividing by zero) of division occurs if logical connectives are interpreted as short-circuit operators in the style of [7, 9].
9. Viewing division as a partial operator is often considered plausible, especially in a computer science context. From the perspective of initial algebra specification of abstract data types that viewpoint is less convincing, however.

   In [4] the construction of some partial meadows, e.g. a meadow where $\frac{1}{0}$ is undefined, is achieved by removing all pairs (p, q) with $p \cdot q \neq 1$ from the graph of a (total) meadow. Remarkably this construction of a well-known partial algebra via a less familiar total algebra seems to provide the simplest construction (by way of an abstract data type specification) of the expansion of the field of rational numbers with a partial division operator.
10. Tuplix calculus ([10]) is a formalism based on meadows aiming at the modular representation of tabular information. In [5] an application of tuplix calculus is developed for the formalization of accumulated interest.

4.3 Common meadows

As mentioned above in [4] partial meadows (using inversive notation) can be found from a meadow by removing all pairs (p, q) with $p \cdot q \neq 1$ from the graph of the inverse operator. I will focus on the simpler case when the underlying meadow is a cancelation meadow (satisfies $x \neq 0 \rightarrow x \cdot x^{-1} = 1$), in which case only the pair $(0, 0)$ needs to be removed from the graph of the inverse function.

A cancellation meadow may be viewed as a zero-totalized (that is made into a total algebra by means of adding zero as the value of inverse on zero) partial cancellation meadow. Instead of zero-totalized partial cancellation meadows one may also consider cancellation meadows that are made total by means of taking an other value for the result of DbZ. A candidate for that value is 1 or any other element of the partial cancellation meadow at hand. It seems that among the existing values 0 is preferable because it leads to the best equational theory. Important is the case where an additional value, say named **a**, is introduced outside the know domain. In data type theory **a** may be termed an error and a preferred notation might be $\perp$, but it seems that there is no need for a "negative" labeling of the role of **a**.

To obtain totalization for a partial cancellation meadow M with domain $\|M\|$ an additional value $\mathbf{a_m}$ is introduced which serves as the interpretation of **a**, which is incorporated as a new element of the domain thus obtaining $\|M_e\| = \|M\| \cup \{\mathbf{a_m}\}$, inverse is made total by setting $\|0^{-1}\| = \mathbf{a_m}$ and the operations are extended to the larger domain as follows: $s + \mathbf{a_m} = \mathbf{a_m} + \mathbf{s} = \mathbf{s} \cdot \mathbf{a_m} = \mathbf{a_m} \cdot \mathbf{s} = \mathbf{a_m^{-1}} = \mathbf{a_m}$ for $s \in \|M_e\|$. This construction produces what I will call a common meadow, thus expressing the belief that this structure more closely models common intuitions about DbZ.

Meadows with zero-totalization of inverse may be called conservative meadows because in a first order setting the addition of a zero-totalized inverse constitutes a conservative extension of the theory of fields. Meadows are assumed to be conservative by default, that is without explicitly stating that a meadow is either common or partial, it is assumed to be a conservative meadow.

When aiming at practical applications working with common meadows embedded in a logic with sequential connectives constitutes an important most obvious alternative to the use of a theory of (conservative) meadows (see item 8 of Section 2 above).

One might criticize the use of a new term (meadow) in the setting of an common meadow, which in the eyes of some observers amounts to no more than a conventional abstract data type oriented view of a field. I disagree with that judgement on the following grounds: (i) by speaking in terms of (some brand of) meadows one has fully embraced the importance of syntax and the fact that inverse (or division) is now a function symbol, (ii) that puts an end to any legitimacy of portraying $\frac{1}{0}$ as irrelevant syntax, and (iii) it renders unreasonable every attempt to dismiss questions about such expressions as indications of lacking knowledge or of a deficient mathematical background of those formulating such questions. Given the long tradition of mathematics teaching where (i), (ii), and (iii) were held in low regard I see ample justification for some new jargon.

4.4 Future objective: separating calculation from semantics

I expect that taking division by zero issues seriously, and in particular doing so via the adoption of the syntax of meadows, even without any commitment to further refinements and totalization techniques, will enable modifications to the teaching of elementary mathematics that make it both simpler and more productive. A future perspective for the application of meadows is to develop rules of calculation for elementary mathematics that do not mix syntax and semantics.

This viewpoint requires some explanation. A typical consequence of the standard view on DbZ is that calculation in a field can only be done by those who already know how it works. Indeed writing $\frac{p}{q}$ always requires some insight in the meaning of q, unless one maintains that reasoning is entirely formalized including non-zeroness conditions and the like, an assumption which I consider less realistic.

This interaction (mixing) of syntax and semantics is quite unfortunate in my view. It is also an avoidable interaction because the mechanisms of calculation (in the context of a family of equational axioms) are in principle entirely independent of the preferred models of those axioms: syntax and semantics can be separated, and calculation can be seen as a generic capability to be mastered in advance of any semantic consideration and independently of any thematically oriented and semantically dedicated syntax. In particular acquiring an awareness of calculation with fractions may usefully precede becoming aware of rational numbers and semantic aspects of the rationals.

5 Concluding remarks

This paper provides a description of issues surrounding division by zero and and attempt to justify the solution of those issues along the lines of the theory of meadows. I might add that developing a theory of meadows appears to be an intriguing objective in its own right irrespective of its applications connected with attempts to challenge the standard view on division by zero. Based on the considerations of this paper I conclude that division by zero may be considered harmless, and that setting the resulting value equal to zero may be considered useful, both judgements standing in contrast with what I have called the standard view on division by zero.

A second conclusion, however, is that meadows alone don't resolve the issue. To see this one may notice that the expressions for solving a quadratic equation mentioned in Paragraph 3.1 don't work in the case of $a = 0$. Indeed the risk that division by zero introduces a calculation error remains. To prevent that from happening a second strategy may be used on top of working in meadows (which is primarily done in order to stay within a two-valued logic), that is the application of the relevant division convention in order to exclude that such calculation errors progress through a text. This is the topic of [12].

In order to check the validity of the application of the relevant division convention (Paragraph 4.2 item 8), reading texts and formulae from the perspective of a three valued logic with McCarthy-style sequential primitives is probably helpful. The latter perspective then plays a role in type checking without being vital for the determination of the meaning of well-typed pieces of text.

References

[1] Jan A. Bergstra, Inge Bethke, and Alban Ponse. Cancellation meadows: a generic basis theorem and some applications. *The Computer Journal*, 56(1): 3–14, doi:10.1093/comjnl/bsx147 (2013).

[2] Jan A. Bergstra, Inge Bethke, and Alban Ponse. Equations for formally real meadows. arXiv.org:1310.5011v3 (2014).

[3] J. A. Bergstra, I. Bethke, and P.H. Rodenburg. A propositional logic with 4 values: true, false, divergent, and meaningless. *J. of Applied Non-Classical Logics*, 5(2): 199–218 (1995).

[4] J.A. Bergstra and C.A. Middelburg. Inversive meadows and divisive meadows. *Journal of Applied Logic*, 9(3): 203–220 (2011).

[5] J.A. Bergstra and C.A. Middelburg. Timed tuplix calculus and the Wesseling and van den Bergh equation. *Scientific Annals of Computer Science*, 23(2): 169–190 (2013).

[6] J.A. Bergstra and A. Ponse. Proposition algebra. *ACM Transactions on Computational Logic*, 12 (3), Article 21 (36 pages), 2011.

[7] J.A. Bergstra and A. Ponse. Proposition algebra and short-circuit logic. In F. Arbab and M. Sirjani (eds.), Proceedings of the 4th International Conference on Fundamentals of Software Engineering (FSEN 2011), Tehran, LNCS 7141, pages 15-31, Springer-Verlag, 2012.

[8] Jan A. Bergstra and Alban Ponse. Signed meadow valued probability mass functions. arXiv.org:1307.5173v1 (2013).

[9] Jan A. Bergstra, Alban Ponse, and Daan J.C. Staudt. Short-circuit logic. arXiv.org:1010.3674v4 (2013).

[10] J.A. Bergstra, A. Ponse and M.B. van der Zwaag. Tuplix calculus. *Scientific Annals of Computer Science*, 18: 35–61 (2008).

[11] J.A. Bergstra and J.V. Tucker. The rational numbers as an abstract data type. *Journal of the ACM*, 54 (2), Article 7 (2007).

[12] J.A. Bergstra and J.V. Tucker. Division safe calculation in totalised fields. *Theory of Computing Systems*, 43(3-4), pp 410-424 (2008).

[13] I. Bethke and P.H. Rodenburg. The initial meadows. *J. Symbolic Logic*, 75 (3), 888-895 (2010).

[14] Y. Komori. Free algebras over all fields and pseudo-fields. Report 10, pp. 9-15, Faculty of Science, Shizuoka University (1975).

[15] J. McCarthy. A basis four a mathematical theory of computation. In P. Braffort and D. Hirschberg (eds.) *Computer Programming and Formal Systems, Noth-Holland*, 33-70 (1963).

[16] H. Ono. Equational theories and universal theories of fields. *Journal of the Mathematical Society of Japan*, 35(2), 289-306 (1983).

Electronic Reports Series of section Theory of Computer Science

Within this series the following reports appeared.

[TCS1403] J.A. Bergstra, I. Bethke, and A. Ponse, *Equations for Formally Real Meadows,* section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1402] J.A. Bergstra and W.P. Weijland, *Bitcoin, a Money-like Informational Commodity,* section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1401] J.A. Bergstra, *Bitcoin, een "money-like informational commodity",* section Theory of Computer Science - University of Amsterdam, 2014.

[TCS1301] B. Diertens, *The Refined Function-Behaviour-Structure Framework,* section Theory of Computer Science - University of Amsterdam, 2013.

[TCS1202] B. Diertens, *From Functions to Object-Orientation by Abstraction,* section Theory of Computer Science - University of Amsterdam, 2012.

[TCS1201] B. Diertens, *Concurrent Models for Object Execution*, section Theory of Computer Science - University of Amsterdam, 2012.

[TCS1102] B. Diertens, *Communicating Concurrent Functions,* section Theory of Computer Science - University of Amsterdam, 2011.

[TCS1101] B. Diertens, *Concurrent Models for Function Execution,* section Theory of Computer Science - University of Amsterdam, 2011.

[TCS1001] B. Diertens, *On Object-Orientation,* section Theory of Computer Science - University of Amsterdam, 2010.

Within former series (PRG) the following reports appeared.

[PRG0914] J.A. Bergstra and C.A. Middelburg, *Autosolvability of Halting Problem Instances for Instruction Sequences,* Programming Research Group - University of Amsterdam, 2009.

[PRG0913] J.A. Bergstra and C.A. Middelburg, *Functional Units for Natural Numbers,* Programming Research Group - University of Amsterdam, 2009.

[PRG0912] J.A. Bergstra and C.A. Middelburg, *Instruction Sequence Processing Operators,* Programming Research Group - University of Amsterdam, 2009.

[PRG0911] J.A. Bergstra and C.A. Middelburg, *Partial Komori Fields and Imperative Komori Fields,* Programming Research Group - University of Amsterdam, 2009.

[PRG0910] J.A. Bergstra and C.A. Middelburg, *Indirect Jumps Improve Instruction Sequence Performance,* Programming Research Group - University of Amsterdam, 2009.

[PRG0909] J.A. Bergstra and C.A. Middelburg, *Arithmetical Meadows,* Programming Research Group - University of Amsterdam, 2009.

[PRG0908] B. Diertens, *Software Engineering with Process Algebra: Modelling Client / Server Architecures,* Programming Research Group - University of Amsterdam, 2009.

[PRG0907] J.A. Bergstra and C.A. Middelburg, *Inversive Meadows and Divisive Meadows*, Programming Research Group - University of Amsterdam, 2009.

[PRG0906] J.A. Bergstra and C.A. Middelburg, *Instruction Sequence Notations with Probabilistic Instructions,* Programming Research Group - University of Amsterdam, 2009.

[PRG0905] J.A. Bergstra and C.A. Middelburg, *A Protocol for Instruction Stream Processing,* Programming Research Group - University of Amsterdam, 2009.

[PRG0904] J.A. Bergstra and C.A. Middelburg, *A Process Calculus with Finitary Comprehended Terms*, Programming Research Group - University of Amsterdam, 2009.

[PRG0903] J.A. Bergstra and C.A. Middelburg, *Transmission Protocols for Instruction Streams*, Programming Research Group - University of Amsterdam, 2009.

[PRG0902] J.A. Bergstra and C.A. Middelburg, *Meadow Enriched ACP Process Algebras*, Programming Research Group - University of Amsterdam, 2009.

[PRG0901] J.A. Bergstra and C.A. Middelburg, *Timed Tuplix Calculus and the Wesseling and van den Berg Equation*, Programming Research Group - University of Amsterdam, 2009.

[PRG0814] J.A. Bergstra and C.A. Middelburg, *Instruction Sequences for the Production of Processes*, Programming Research Group - University of Amsterdam, 2008.

[PRG0813] J.A. Bergstra and C.A. Middelburg, *On the Expressiveness of Single-Pass Instruction Sequences*, Programming Research Group - University of Amsterdam, 2008.

[PRG0812] J.A. Bergstra and C.A. Middelburg, *Instruction Sequences and Non-uniform Complexity Theory*, Programming Research Group - University of Amsterdam, 2008.

[PRG0811] D. Staudt, *A Case Study in Software Engineering with PSF: A Domotics Application*, Programming Research Group - University of Amsterdam, 2008.

[PRG0810] J.A. Bergstra and C.A. Middelburg, *Thread Algebra for Poly-Threading*, Programming Research Group - University of Amsterdam, 2008.

[PRG0809] J.A. Bergstra and C.A. Middelburg, *Data Linkage Dynamics with Shedding*, Programming Research Group - University of Amsterdam, 2008.

[PRG0808] B. Diertens, *A Process Algebra Software Engineering Environment*, Programming Research Group - University of Amsterdam, 2008.

[PRG0807] J.A. Bergstra, S. Nolst Trenite, and M.B. van der Zwaag, *Tuplix Calculus Specifications of Financial Transfer Networks*, Programming Research Group - University of Amsterdam, 2008.

[PRG0806] J.A. Bergstra and C.A. Middelburg, *Data Linkage Algebra, Data Linkage Dynamics, and Priority Rewriting*, Programming Research Group - University of Amsterdam, 2008.

[PRG0805] J.A. Bergstra, S. Nolst Trenite, and M.B. van der Zwaag, *UvA Budget Allocatie Model*, Programming Research Group - University of Amsterdam, 2008.

[PRG0804] J.A. Bergstra and C.A. Middelburg, *Thread Algebra for Sequential Poly-Threading*, Programming Research Group - University of Amsterdam, 2008.

[PRG0803] J.A. Bergstra and C.A. Middelburg, *Thread Extraction for Polyadic Instruction Sequences*, Programming Research Group - University of Amsterdam, 2008.

[PRG0802] A. Barros and T. Hou, *A Constructive Version of AIP Revisited*, Programming Research Group - University of Amsterdam, 2008.

[PRG0801] J.A. Bergstra and C.A. Middelburg, *Programming an Interpreter Using Molecular Dynamics*, Programming Research Group - University of Amsterdam, 2008.

The above reports and more are available through the website: www.science.uva.nl/research/prog/

Electronic Report Series

section Theory of Computer Science
Faculty of Science
University of Amsterdam

Science Park 904
1098 XG Amsterdam
the Netherlands

www.science.uva.nl/research/prog/