Journal of Universal Computer Science, vol. 30, no. 3 (2024), 289-307
submitted: 27/5/2023, accepted: 18/10/2023, appeared: 28/3/2024

Synthetic Fracterm Calculus

Jan Bergstra
(University of Amsterdam, The Netherlands
https://orcid.org/0000-0003-2492-506X, j.a.bergstra@uva.nl, janaldertb@gmail.com)

John V. Tucker
(University of Swansea, Swansea, Wales, United Kingdom
https://orcid.org/0000-0003-4689-8760, j.v.tucker@swansea.ac.uk)

Abstract: Previously, in [Bergstra and Tucker 2023], we provided a systematic description of elementary arithmetic concerning addition, multiplication, subtraction and division *as it is practiced*. Called the *naive fracterm calculus*, it captured a consensus on what ideas and options were widely accepted, rejected or varied according to taste. We contrasted this state of the practical art with a plurality of its formal algebraic and logical axiomatisations, some of which were motivated by computer arithmetic. We identified a significant gap between the wide embrace of the naive fracterm calculus and the narrow precisely defined formalisations. In this paper, we introduce a new intermediate and informal axiomatisation of elementary arithmetic to bridge that gap; it is called the *synthetic fracterm calculus*. Compared with naive fracterm calculus, the synthetic fracterm calculus is more systematic, resolves several ambiguities and prepares for reasoning underpinned by logic; indeed, it admits direct formalisations, which the naive fracterm calculus does not. The methods of these papers may have wider application, wherever formalisations are needed to analyse and standardise practices.

Keywords: fracterm calculus, partial meadow, common meadow, abstract data type
Categories: F.4.2, F.4.1, D.3.1
DOI: 10.3897/jucs.107082

1 Introduction

Elementary arithmetic is commonly thought of as the mathematics of addition, subtraction, multiplication and division as it has been – and continues to be – taught in schools and practiced in everyday life for centuries. A remarkable textbook tradition has recorded its development and, through exercises based on practical problems, has witnessed and recorded historical change [Morgan 2023]. Of course, the mathematics itself has been largely settled since the 16th Century with 'modern' symbols, formulae, equations and rules. However, like most things in human life, teaching and even understanding elementary arithmetic is not entirely professionally agreed and stable, let alone uniform in individual national traditions: one need only think of fractions and division by zero.

In [Bergstra and Tucker 2023], motivated by problems of division by zero in computing and fractions in mathematics education, we constructed a systematic informal description of a consensus on thinking about elementary arithmetic. We made explicit what ideas and options were widely accepted, rejected or simply varied according to particular perspectives or personal taste. We called this description the *naive fracterm calculus* (NFTC). In [Bergstra and Tucker 2023], we also contrasted this 'state of the practical art' with a plurality of formal algebraic and logical axiomatic counterparts, most of which were motivated by the semantics of computer arithmetic. Our naive fracterm calculus included so much flexibility that it could generate a range of informal requirements analysis on behalf of arithmetical practitioners. Thus, we identified a significant gap between the wide embrace of the naive fracterm calculus and the tight grip of a precisely defined formal calculi.

In this paper, we introduce a new informal axiomatisation of elementary arithmetic to bridge that gap; it is called the *synthetic fracterm calculus* (SFTC). Compared with naive fracterm calculus, the synthetic fracterm calculus is more systematic, resolves ambiguities allowed in the naive calculus, and prepares for reasoning underpinned by logic. Indeed, it is more like a requirements analysis, in that it invites direct formalisations, which naive fracterm calculus does not, because of its many acceptable options. The end users of the synthetic fracterm calculus that we have in mind include teachers of elementary arithmetic.

Since the synthetic fracterm calculus will reference the naive fracterm calculus, to read this paper, knowledge of [Bergstra and Tucker 2023] is necessary, though we will need to repeat material from that paper to show the connections. First, we will summarise some of the special features of our methodology.

1.1 On informal and formal calculi

Practical arithmetic, even at the elementary levels of schools, is not easy to discuss in an explicit and systematic way – as an extensive literature on teaching arithmetic largely confirms. The term 'practical' refers to practices by people, which are inherently diverse and can be incomparable.

By 'calculus' we have in mind operations, expressions, equations, and methods for their application as found in school textbooks; note that a calculus is *not* necessarily a formal object.[1] Thus, our task in SFTC, like in NFTC, is anthropomorphic as well as mathematical.

At the outset, in [Bergstra and Tucker 2023], we conceived of the task in these four stages:

1. *Raw Arithmetic.* Examine arithmetical practices 'in the wild'.

2. *Naive Arithmetic.* Formulate a description of what seems to be a consensus on what ideas and conventions are agreed, disputed or remain ambiguous in peoples' practices.

3. *Synthetic Arithmetic.* Tighten and refine the informal description that is Naive Arithmetic to resolve ambiguities and arbitrate some disputed ideas and conventions and make it fit for systematic logical reasoning.

4. *Formal Calculi.* Propose formal calculi for elementary arithmetic based on the informal analysis of the Naive and Synthetic.

[1] In mathematical logic and computing, calculi are formal systems with carefully defined notations, rules and meanings capable of metamathematical investigation and machine implementations. It's nice to remember the etymology of 'calculus' in Roman literature: pebble or stone used in arithmetic reckoning on counting boards.

In this paper, we focus on building a calculus that uses Naive Arithmetic to create an intermediate Synthetic Arithmetic, as a requirements analysis for a formalisation.

The technical needs of division require special attention to fractions. The explorations of Raw Arithmetic reveal considerable conceptual difficulties with this term. The issue is the range of meanings implied by the use of the word ‘fraction’. Already in formulating the all embracing description of Naive Arithmetic, we decided to highlight the issue by coining and using the word ‘fracterm’ instead (see section 3.1 below). Thus, our use of the words *fracterm calculus* of the titles is an attempt at an informal analysis of peoples’ practices when actually doing arithmetic, especially divisions.

Fracterms *and* fractions can be used in Naive Arithmetic, but in the more defined SFTC, ‘fraction’ is banished from this more precise description of Synthetic Arithmetic.

The formal calculi come from studies of computer arithmetic as abstract data types. We have studied some 5 basic calculi, primarily distinguished by decisions about the meaning of $\frac{1}{0}$. Our SFTC is a refinement of the consensus NFTC that informally captures properties that our formalisations respect. Of these we highlight two semantic decisions: the case when $\frac{1}{0}$ does not exist – the expected school orthodoxy – and the case when $\frac{1}{0}$ is an error flag – the common semantics of school calculators.

1.2 Methodology

To design the SFTC we use the same methods we employed in making the informal NFTC. In summary, guiding our design are ideas about abstract data types for arithmetic, which provide: intuitions and a means to explore what elementary school arithmetic is about; what is essential in its use; and how arithmetic can be organised. The theory of abstract data types offers an approach to the description of arithmetic with precise ideas and tools for working with numbers, operations, expressions and equations centre stage – just as in elementary arithmetic.

The new informal synthetic calculi we propose here improve on arithmetic as expressed in natural language, but remain informal and become a reference for the design of formal calculi of the kind also on display in [Bergstra and Tucker 2023].

Our methodology makes explicit our objectives and claims about potential users, our design options and our design decisions. What drives the development are arguments and perceptions of *meaningfulness* or *plausibility* of decisions, and their *relevance* to users. We use the intuitions of abstract data type theory to source and choose technical options to adopt, reject, or leave open. For this, we have a terminology: options are *committed*, *adversely committed*, or *uncommitted*, respectively.

1.3 Structure and contributions of the paper

The contributions of this paper are:

1. To design a new informal calculus, named synthetic fracterm calculus (SFTC) for arithmetic focussed on division. SFTC expresses a *philosophy of formality* while leaving aside actual formalisations with their many details. Thus, SFTC aims at greater conceptual clarity and ease of practical use than any of its formalisations could offer.

2. To make explicit our methods in developing the calculus – levels of description in the process of making formalisations of established practices, and objectives, options and decisions for and against technical features.

3. To discuss and compare some formal calculi for arithmetic with division.

In Section 2, we clarify why we wish to pursue arithmetic at this elementary and fundamental level. In Section 3, we introduce the basics of fracterms and our synthetic fracterm calculi SFTC. In Section 4, we develop a more comprehensive list of commitments and non-commitments for NFTC. In Section 5, we look at rewriting expressions in NFTC. Then we turn our attention to the formal calculi. In Section 6, we look at formal calculi for meadows, including common meadows in which division is made total by an 'error' flag $0^{-1} = \bot$ and involutive meadows in which $0^{-1} = 0$. In Section 7, we conclude with some reflections, including on the scope of the four-stage methodology.

To understand this paper fully, the reader will need to refer to our [Bergstra and Tucker 2023] in several places.

2 Why synthetic fracterm calculi?

As we discussed in [Bergstra and Tucker 2023], an aversion to, and dismissal of, formalisation seems to be part of the consensus on elementary arithmetical practice. Further, a calculus, whether formal or informal, is unlikely to be an attractive point of departure for teaching, as arithmetic is so rooted in everyday affairs. However, we believe that *the value of a calculus to a teacher is that it can bring to the surface insights hidden in practice and so can contribute to his or her mastery of the subject matter*.

In [Bergstra and Tucker 2023], a list of claims were made about elementary arithmetic that taken together was used as a package of attitudes and views with which to design a naive fracterm calculus. Let us begin with recalling some of these claims before turning to the claims for a synthetic fracterm calculus.

2.1 Abstract data types: formal versus informal

The demands of computer applications have required a deep investigation of the nature of data and what makes abstract models of data useful for programming. The theory of abstract data types provides a comprehensive algebraic and logical understanding of data whose central concepts and methods can be easily formalised and expressed in software. These concepts are the basis of our approach to elementary arithmetic. However, in [Bergstra and Tucker 2023], we began our preparations for the naive fracterm calculus with this caution:

Claim 2.1. *Once aware of the notion of a data type, most practitioners of elementary arithmetic prefer not to understand the well-known portfolio of arithmetics (naturals, integers, rationals, etc) as a portfolio of data types. In particular, having to think in terms of signatures and syntax will create resistance.*

With naive fracterm calculus we used the intuitions of arithmetical data types to detect, analyse and highlight practices. With synthetic fracterm calculus (SFTC) we intend to preserve many more of the intuitions of arithmetical data types, while being unconstrained – and, in fact, not bothered – by the details of formalization. Thus, SFTC replaces a simplistic and thereby somewhat overly optimistic picture of arithmetic with a simplified, yet instantly recognisable, sharper picture. For instance, in SFTC we will work as if it were obvious that numbers and expressions are different entities even if we know that in some cases telling both categories of entities apart is far from straightforward. In other words, in principle, SFTC adopts an implicit distinction between syntax and semantics, which NFTC does not.

In [Bergstra and Tucker 2023] we have formulated the following aim for NFTC.

Design Objective 2.1. *Adopting the tenets of NFTC constitutes a majority position among persons interested in elementary mathematics, including the majority of school teachers.*

For SFTC we expect a longer distance from the attitudes of professionals in arithmetical practice.

2.2 Basic Claims of NFTC and SFTC

Recall from [Bergstra and Tucker 2023], in adopting NFTC, one endorses these claims:

Claim 2.2. *Arithmetic is based in natural language, taking the form of an extension of core natural language with certain notations and conventions. Arithmetic is therefore conceptually prior to logic, and therefore there is no need to look for a logical basis of arithmetic. Any proposal for a logical basis for arithmetic will turn out to be both defective and artificial.*

For the acquisition of basic competence in a natural language awareness of grammar is not essential. This idea may be extended to a claim about arithmetic as follows:

Claim 2.3. *Arithmetic is conceptually prior to syntax, and therefore there is no need to look for a syntactic basis of arithmetic. Any proposal for a syntactic basis for arithmetic will turn out to be both defective and artificial.*

Claim 2.4. *Logical complications do not stand in the way of a proper understanding of arithmetic, even if some logical complications may prove rather hard to settle.*

Claim 2.5. *Elementary arithmetic is not protected against inconsistency and error on the basis of a package of assumptions and reasoning patterns which has been deliberately designed to stay away from inconsistency and error as much as possible while still covering sufficient ground. On the contrary, elementary arithmetic acquires its stability and reliability from the daily practice of a community of users who will eventually correct one another if needed.*

The following aim, and corresponding claim, point to the advantages that might result from adopting SFTC as a refinement of NFTC.

Design Objective 2.2. *Our SFTC approach to arithmetic is designed to enable the basics of arithmetic become subject to logical reasoning at least as a matter of prinicple. Thus, we suppose that to an adopter of SFTC the priority and overarching significance of logical reasoning is acknowledged and familiar.*

Claim 2.6. *Because the SFTC approach to arithmetic subjects the basics of arithmetic to logical reasoning – unlike NFTC – adopting SFTC turns arithmetic into a world where logical reasoning can be amply applied and can be systematically exercised.*

3 Fracterms and synthetic fracterm calculi

Both NFTC and SFTC are supposed to be informal stand-alone perspectives on elementary arithmetic that are not in need of any foundations provided by additional formalisation.

3.1 Fracterms

As argued in [Bergstra 2020], 'fraction' is an ambiguous notion. The meaning of fraction ranges across various forms of expressions, values and rational numbers. Remarkably, there is no obvious majority position in this spectrum of interpretations of fraction. In NFTC, introducing the word 'fracterm' was an early intervention to limit the ambiguity of fraction by always opting to mean an expression when a distinction between expression and value is needed. Complementary to fracterm, 'quotient' restricts the ambiguity of fraction by opting for a value when a distinction between value and expression is needed. Where fracterm has a syntactic bias, quotient has a complementary semantic bias. Both fracterm and quotient are words inherited from NTFC for use in SFTC.

Definition 3.1. *A fracterm is a structured entity involving a numerator, a denominator and a leading function symbol for division.*

Definition 3.2. *A quotient is the result of performing division on a pair of arguments. The respective arguments of a quotient are sometimes called dividend and divisor.*

A fracterm comprises the inputs of a division, irrespective of whether or not the operation can actually be performed. A quotient, on the other hand, specifies the output of a division; a quotient implies that division must have been enabled. However:

Design Choice 3.1. *Unlike in NFTC, quotient and fracterm are distinguished in SFTC.*

In NFTC we also defined sums and sumterms, and products and producterms, but made no use of them; similarly:

Design Choice 3.2. *In the context of SFTC sum and sumterm are not distinguished, and product and producterm, are not distinguished. Both 'sumterm' and 'producterm' are not used in SFTC.*

As in NFTC, the reason to treat division differently from addition and multiplication is that fracterms play a special role in elementary arithmetic.

3.2 Synthetic Fracterm Calculus (SFTC)

SFTC comes about from the following combination of design objectives, design choices, and underlying more general definitions.

Design Objective 3.1. *Many judgements of elementary arithmetic are uncontroversial. These judgements belong to NFTC and are inherited by SFTC; in particular, SFTC adopts all arithmetical truths as captured by equations in NFTC.*

NFTC focuses on the large body of uncontroversial assertions in and about elementary arithmetic.

Design Objective 3.2. *While NFTC is 'by design' immune against being split into a plurality of views on details, SFTC incorporates philosophical positions in debates about the logic and meaning of unusual arithmetical ideas and statements.*

Consider a non-denoting expression that is not allowed in NFTC, such as $\frac{1}{0}$ or the equation $\frac{1}{0} = \frac{1}{0}$.

Design Choice 3.3. *SFTC is committed to accepting $\frac{1}{0}$ as a fracterm, but considers it an 'unusual' fracterm, and preferably unusual fracterms should not be used in written or spoken language about elementary arithmetic.*

The description 'let $g(x) = \frac{1}{x}$ with $x > 0$' is acceptable, although reading from left to right the fragment $\frac{1}{x}$ is at risk of being non-denoting when first read. The occurrence of $\frac{1}{x}$ is potentially unusual but not necessarily unusual, and in an adequate context (such as the context $x > 0$) it is acceptable.

Design Choice 3.4. *SFTC is uncommitted to a judgement of the validity of $\frac{1}{0} = \frac{1}{0}$. However, unlike NFTC, SFTC is committed to reflection about the validity of the 'unusually' formulated statement $\frac{1}{0} = \frac{1}{0}$, and acknowledges that different answers are possible depending on the context at hand.*

Thus, questions about such expressions are complicated and lead to questions about logic.

Design Choice 3.5. *Unlike NFTC, SFTC is committed to two-valued classical logic in all but unusual cases.*

Design Choice 3.6. *Like NFTC, SFTC makes use of the word 'fracterm' where readers might expect 'fraction'.*

Fracterm is used in SFTC in such a way that it acquires the same meaning as "fractional expression" that can be found in some school arithmetics. SFTC provides not even a marginal role for the word 'fraction' and is *opposed* to the use of it.

Definition 3.3. *In the context of SFTC, 'fraction' is claimed to be ambiguous in the following manner: fracterm can both refer to a fracterm or to the value of a fracterm (and perhaps also to notions concerning fracterms and their abstractions).*

Thus, in view of Definition 3.3, while in NFTC fracterm and fraction are the same notion, in SFTC that is not the case. As we prefer not to constrain this work by any claim about the ontology of fractions – that being a difficult subject – we will use fracterm at the expense of 'fraction'.

Design Objective 3.3. *Unlike NFTC, SFTC is committed to the belief that $\frac{1}{2}$ and $\frac{2}{4}$ are different fracterms, while acknowledging that both denote the same quotients. Like NFTC, SFTC is committed to the belief that $\frac{1}{2}$ is a simplification of $\frac{2}{4}$.*

Design Objective 3.4. *Like NFTC, STFC is committed to the validity of*

$$\phi(x) \equiv (x \neq 0 \rightarrow \frac{x}{x} = 1).$$

The latter assertion is true with the 'short-circuit reading' of implication: if the condition fails (i.e., if $x = 0$) then $\phi(x)$ is considered valid and no attention is paid to the conclusion at all.

The short circuit semantics of logical connectives plays a role in our development and we adopt a notation to make it explicit when needed. In the following claim we use the notation $\circ\!\!\rightarrow$ for short-circuit implication; similar notations exist for all the connectives, e.g., ${}_{\circ}\!\wedge$, ${}^{\circ}\!\vee$, $\wedge\!{}_{\circ}$, $\vee\!{}^{\circ}$ and so on [Bergstra et al. 1995].

Design Choice 3.7. *Like NFTC, SFTC acknowledges that short-circuit implication plays a central role so that it may warrant its own notation (for which no generally agreed candidate is available) and then positively affirms that:*

$$x \neq 0 \circ\!\!\rightarrow \frac{x}{x} = 1.$$

SFTC acknowledges (or at least does not reject) differences in plausibility between various assertions each of which is neither clearly true nor clearly false.

Unlike NFTC, SFTC accepts the sentence '$\frac{1}{0}$ is a fracterm', while at the same time acknowledging that the semantics of fracterm is open for debate. What can be said and written in SFTC instead is that 'dividing 1 by 0 is *unusual* because it will not deliver a well-specified result'. An important practical working principle is this:

Existence in SFTC is understood as meaningful existence, but perhaps in some extended framework.

Design Choice 3.8. *Like NFTC, SFTC is committed to the existence of various kinds of numbers: natural, integer, rational, real and complex.*

However, with some caution as to the real and complex numbers:

Design Choice 3.9. *Like NFTC, SFTC is adversely committed to attempts to define the various numbers in more detail, with the idea that these are self-supporting intuitions that come about from natural language as much as from any ideology on how to design mathematical theory and the corresponding notational conventions and patterns of reasoning.*

The question 'is division a function name just as addition' is affirmed in SFTC:

Design Choice 3.10. *Unlike NFTC, SFTC is committed to the idea that naturals, integers, and rationals are domains of data types that in turn serve as plausible representatives of corresponding abstract data types. However, the notion of a signature is not made explicit in SFTC.*

4 More commitments and non-commitments of SFTC

The objectives and choices mentioned above may be understood as desiderata for SFTC, requiring that SFTC should satisfy certain properties. Again, we think of SFTC as a package of opinions that are shared by some people, though we need not suppose supporters of SFTC are happy with NFTC. Unlike for NFTC, consistency of the package of opinions is now a major concern. This is because a rationale for the package of views that constitutes SFTC is to allow us to justify arithmetical judgements on the basis of logic and axioms.

Someone who adopts SFTC will have opinions about many issues that will need to be established. Opinions about technical options will need to be accepted, rejected or left open. We will progress the design of SFTC with a list of options that we are not committed to and then with a list of options we are committed to. These listings are not meant to be complete in any rigorous sense. However, they follow closely a similar set of issues and options that were accepted, rejected or open in the creation of NFTC.

The pattern of decisions for SFTC that follows is based on revisiting the decisions for NFTC listed in Section 4 of [Bergstra and Tucker 2023]. Many of the commitments of NFTC are inherited by SFTC.

4.1 Non-commitments of SFTC

Issues to do with syntax and semantics

1. Unlike in NFTC, an informal distinction between syntax and semantics is present in SFTC.
2. Like NFTC, SFTC is uncommitted to the distinction between a value and an expression for numerals, e.g., 251 can be an expression as well as a value.
3. Like NFTC, SFTC is uncommitted to any detailed ontology that might be involved in a distinction between syntax and semantics. However, if for some concept, say fraction, a distinction between syntax and semantics is put forward then the concept will become ambiguous and the name of the concept will be a candidate for being replaced by less ambiguous terminology. Resolution of ambiguities is considered a matter of conceptual analysis of arithmetical thought and so a matter of natural language processing, and not to be considered a mathematical subject.
4. Unlike in NFTC, in SFTC fracterm is considered a mathematical notion (following [Fandino Pinilla 2007]).
5. Like in NFTC, for identities and formulas with free variables, the scope of the variables depends on the context, which is defined in natural language.

Issues to do with logical formulae

For NFTC we also gave four non-commitments about formulae in [Bergstra and Tucker 2023]. They concern the (i) meaning of quantification, (ii) absence of formal logic theory, and (iii) absence of foundational mathematical theories (such as set theory, constructivism). All these tenets are carried over in SFTC.

4.2 Commitments of NFTC

The following list of commitments further details the view of arithmetic that NFTC is supposed to comprise.

Issues to do with algebra and calculation

1. In SFTC, fracterm is considered a mathematically relevant notion, which, however, need not be rigorously defined as if it were a proper mathematical concept; fracterm is like proof, definition, theorem, which are notions belonging to mathematical practice rather than to mathematical substance and which, at least working at the informal level of SFTC, can be left without proper mathematical definitions.
2. Like NFTC, SFTC is committed to the existence of a bundle of widely agreed upon closed identities of the form $t = r$, as well as to a bundle of universally agreed upon inequations of the form $t \neq r$.

   An informal theory is supposed available (e.g., in the form of a plurality of algorithms) concerning how to distinguish between $t = r$ and $t \neq r$ for closed expressions t and r.

3. Like in NFTC, in SFTC variables in an equation may be implicitly universally quantified, in which case one may speak of a law regarding such variables. Alternatively, a variable may be supposed to have a fixed, but arbitrary value in some domain. The latter form of quantification is often used in formulating and proving results.

   However, in an equation with parameters (e.g., $a \cdot x^2 + b \cdot x + c = 0$), one assumes that at some level a, b, c are universally quantified, while x denotes one or more specific values depending on a, b, c. In the latter case no universal quantification over x is implied, rather x plays the role of an additional constant which is further specified by the mentioned equation.

   The natural language, in which formulas are embedded, determines the use of a variable.

Issues to do with arithmetic and calculation

4. Elementary arithmetic is primarily embedded in natural language for which 100% precision is both unachievable and unnecessary. Like NFTC, there is in practice only marginal disagreement about the status of facts in the language of SFTC.

5. Like in NFTC, in SFTC calculation is understood as a mechanical procedure. Unike in NFTC, calculation is also understood as an instance of mathematical or logical reasoning, though calculating efficiently is considered to be an independent competence.

6. SFTC maintains a commitment to calculation with decimal naturals and decimal integers.

7. Defining expressions in general requires an inductive definition and comes with the ability to count sizes of an expression. Unlike for NFTC, for SFTC it is plausible to consider the notion of an expression to be prior to counting.

   However, admittedly any attempt to define the class of expressions with full precision then will be confronted with marginal cases which may uncover a certain lack of precision of the definition.

8. SFTC acknowledges the relevance of a systematic notion of legality for expressions, though does not impose any particular outcomes on that endeavour [Bergstra and Tucker 2023].

Issues to do with relevance

9. Unlike for NFTC, when adopting SFTC, for each expression or equation, the first question is about its *meaning* and not about its *relevance*.

10. Unlike NFTC, SFTC rates truth and consistency over relevance in the following sense: only once matters of meaning, truth, and consistency have been established in a satisfactory manner, questions regarding potential relevance matter.

11. Like NFTC, SFTC does not come with axioms and a proof system for relevance, though there are some rules of thumb.

5 Rewriting expressions in SFTC

Like for NFTC, knowledge regarding SFTC comes about from experience – inductively rather than deductively. Once someone has become convinced that, say,

$$\frac{5 \cdot 2}{7 \cdot 2} = \frac{5}{7}, \frac{8 \cdot 3}{13 \cdot 3} = \frac{8}{13}, \frac{1 \cdot 5}{15 \cdot 5} = \frac{1}{15}$$

it becomes plausible to guess that for all x, y and positive natural n,

$$\frac{x \cdot n}{y \cdot n} \text{ may be replaced by } \frac{x}{y}.$$

Such observations suggest underlining much of the calculating practices of elementary arithmetic are particular forms of *term rewriting* guided by *equational rules*.

Definition 5.1. *Given a collection of rewrite rules R, $t \Rightarrow_R^\star s$ asserts that with 0 or more steps from R, t can be rewritten to s. In particular, if t has been obtained using the rules in R then also s can be obtained and $t = s$ holds.*

5.1 Rewriting by example

Substitutions into equations and term rewriting are not quite the same. For instance,

$$\frac{x \cdot 7}{y \cdot 7} \Rightarrow \frac{x}{y}$$

is a valid rewrite rule for SFTC while the corresponding equation

$$\frac{x \cdot 7}{y \cdot 7} = \frac{x}{y}$$

fails because $y = 0$ has not been excluded!

Applying the rule $\frac{x \cdot 7}{y \cdot 7} \Rightarrow \frac{x}{y}$ however, is only enabled if for some r, s, a fracterm $\frac{r \cdot 7}{s \cdot 7}$ has been obtained from which it follows that $s \neq 0$. It then follows that as a rewrite rule

$$\frac{x \cdot 7}{y \cdot 7} \Rightarrow \frac{x}{y}$$

is correct without any need to require that $y \neq 0$.

Many more rules can be proposed. For instance:

$$\frac{x \cdot y}{u \cdot v} \Rightarrow \frac{x}{u} \cdot \frac{y}{v} \text{ and its converse } \frac{x}{u} \cdot \frac{y}{v} \Rightarrow \frac{x \cdot y}{u \cdot v}.$$

In SFTC it is considered unproblematic to write $t = r$ if $t \Rightarrow r$ is actually meant. The cost of this latter convention is significant however, as rewriting collides with commutativity of $=$. Indeed, while $\frac{x \cdot z}{y \cdot z} \Rightarrow \frac{x}{y}$ is valid, $\frac{x}{y} \Rightarrow \frac{x \cdot z}{y \cdot z}$ fails for $z = 0$, so that dropping the orientation of arrows is unwarranted in this case.

Division by 1 takes the form of the rule $\frac{x}{1} \Rightarrow x$. For numerator 0 one finds the rule $\frac{0}{x} \Rightarrow 0$. Now, taking 0 for x will be considered an unusual case for SFTC, as it involves contemplating a term $\frac{0}{0}$. SFTC acknowledges that a spectrum of different options is available to extend the core of SFTC with perspectives on division by zero.

Our comments about expressions in NFTC carry over; in particular, we recall:

Definition 5.2. *An expression is fracfree if it has no fracterm as a subterm. A flat fracterm is a fracterm of the form $\frac{p}{q}$ with p and q both fracfree.*

It is immediate that fracfree expressions are closed under addition, subtraction and multiplication. The familiar rule

$$\frac{x}{y} + \frac{u}{v} \Rightarrow \frac{x \cdot v + u \cdot y}{y \cdot v}$$

allows the following conclusion: *a sum of flat fracterms can be replaced by a single flat fracterm (with the same value)*.

6 Formalisations of SFTC

In [Bergstra and Tucker 2023], we contrasted NFTC, our description of Raw Arithmetic, with five very different formalisations of arithmetics. These formalisations were based on equational specifications of different computer arithmetics, viewed as abstract data types. Their technical focus on equations and equational term rewriting for calculation and reasoning, together with a close relationship with rational numbers via initial algebra semantics, fit well with the *formal* study of Raw Arithmetic and its informal descriptions of Naive and Synthetic Arithmetics, NFTC and SFTC, respectively.

Although this investigation belongs to computer science, we will contrast SFTC with some formalisations that analyse semantical ideas that can be encountered in the schoolroom.

The first two we consider are formalisations of semantic behaviours that can be found in pocket calculators. The key point is that in calculators, and all computing systems, arithmetic operations, including division, must return a value of some kind that the user can interpret. Actually, there is a number of responses to typing $\frac{1}{0}$ into calculators, which we will not catalogue here. A simple distinction is whether $\frac{1}{0}$

(i) stops the calculation, or

(ii) allows calculation to continue, albeit with some 'red flag' to say something is not right.

Most calculators satisfy (i) through a variety of means. Today's common semantics in calculators takes $\frac{1}{0}$ to be an error flag. A now less common semantics – but one that can be still found in new calculators – takes $\frac{1}{0} = 0$. This convention starts early : for example, the (1980 onwards) Japanese mass-produced Casio Electronic Calculator HL-809 gave $\frac{1}{0} = 0$, though with a tiny 'E' displayed elsewhere in its LCD.

The third formalisation is of the common orthodoxy in teaching: $\frac{1}{0}$ does not exist.

Now, partiality is important in computing, not least because non-terminating computations are common. Certainly, division of x by y is computed by repeated subtraction of y from x, which typically fails to terminate if $y = 0$ is allowed.

Although partiality has no place in implementations, especially of data types, it plays a valuable role in specifications. For example, partiality is an important semantic feature of the specification language CASL [Bidoit and Mosses 2004]. There are different interpretations of partiality in addition to the orthodox 'no element' of the schoolroom:

partiality stands for some element yet to be defined, possibly one that is quite arbitrary. The Suppes-Ono semantics can be considered to be an instance where there seems to be a natural choice of element, i.e., 0.

6.1 Equations for common meadows

The simple idea that errors propagate can be given an algebraic expression: a flag $\perp$ is added to the numbers that is *absorbtive*: if $\perp$ is an argument to an operator then the result returned is $\perp$. So, e.g.:

$$x + \perp = \perp \text{ and } x \cdot \perp = \perp.$$

Thus, $\perp$ propagates like an error message and closes down calculation.

A common meadow is a field F whose operations $x + y, -x, x \cdot y$ are expanded by x/y, and whose constants $0, 1$ are enriched by $\perp$, and wherein

$$\frac{1}{0} = \perp.$$

The formal fracterm calculi of common meadows – as discussed in [Bergstra and Ponse 2021] and in [Bergstra and Ponse 2016] – have many different axiomatisations. Table 1 lists equations following the presentation of [Bergstra and Tucker 2023b], though with some minor modifications.

The axioms for common meadows in Table 1 allow *fracterm flattening*: each fracterm can be proven equal to a flat fracterm, where a fracterm is *flat* if it contains precisely one occurrence of the division operator (i.e., the top level occurrence). (Recall Definition 5.2.)

6.2 Equations for involutive meadows

For many situations using

$$\frac{1}{0} = 0.$$

is convenient and common practice, and has been chosen for a long time. It is used widely when one wants to dodge partiality: in today's theorem provers, such as Coq and Lean, and in logic (such as the model theory of fields); and it is still used in some contemporary calculators. It is certainly a workable mathematical convention that can be used (e.g., [Bergstra 2019]).

This semantics for total division is that of involutive meadows, which we will also refer to as *Suppes-Ono* meadows.

An involutive meadow is a field F whose operations $x + y, -x, x \cdot y$ are expanded by x/y which is made total by adopting $\frac{1}{0} = 0$. The name Suppes-Ono fracterm calculus is motivated by [Suppes1975] and [Anderson and Bergstra 2020] and the observation that [Ono 1983] gave the first significant analysis of the logical consequences of adopting $1/0 = 0$.

Models of Suppes-Ono FTC are called involutive meadows because inverse is an involution $(x^{-1})^{-1} = x$ (see [Bergstra et al. 2009] and [Bergstra and Middelburg 2011]. Meadows of rational numbers are an early prime example (see [Bergstra and Tucker 2007]). Equations of a fracterm calculus for involutive meadows are listed in Table 2.

Surprisingly, in Suppes-Ono FTC, fracterm flattening *fails*, as shown in [Bergstra and Middelburg 2016]. In Suppes-Ono FTC, according to [Bergstra et al. 2013], fracterms

$$
\begin{aligned}
(x+y)+z &= x+(y+z)\\
x+y &= y+x\\
x+0 &= x\\
x+(-x) &= 0\cdot x\\
x\cdot(y\cdot z) &= (x\cdot y)\cdot z\\
x\cdot y &= y\cdot x\\
1\cdot x &= x\\
x\cdot(y+z) &= (x\cdot y)+(x\cdot z)\\
-(-x) &= x\\
0\cdot(x+y) &= 0\cdot(x\cdot y)\\
x+\bot &= \bot
\end{aligned}
$$

$$
\begin{aligned}
\frac{x}{1} &= x\\
-\frac{x}{y} &= \frac{-x}{y}\\
\frac{x}{y}\cdot\frac{u}{v} &= \frac{x\cdot u}{y\cdot v}\\
\frac{x}{y}+\frac{u}{v} &= \frac{(x\cdot v)+(y\cdot u)}{y\cdot v}\\
\frac{x}{y+0\cdot z} &= \frac{x+0\cdot z}{y}\\
\bot &= \frac{1}{0}
\end{aligned}
$$

Table 1: Equations for a fracterm calculus for common meadows

can be rewritten to *sums* of flat fracterms. Theoretical work on Suppes-Ono FTC can be found in [Bethke and Rodenburg 2010, Bethke et al. 2015]. Suppes-Ono fracterm calculus is called Division by Zero Calculus in [Michikawi et al. 2016, Okumura 2018].

6.3 An orthodox mathematical interpretation of SFTC

Finally, we turn to the orthodoxy that $\frac{1}{0}$ simply does not exist. Already in [Bergstra and Tucker 2023], we made the following claim.

Claim 6.1. *If a formal basis for arithmetic is to be adopted then a person who is initially disinclined to adopt any formalisation of elementary arithmetic, is likely to prefer an fracterm calculus presenting division as a partial function over any of the plurality of calculi that incorporate division as a total function.*

$$(x + y) + z = x + (y + z) \tag{1}$$
$$x + y = y + x \tag{2}$$
$$x + 0 = x \tag{3}$$
$$x + (-x) = 0 \tag{4}$$
$$x \cdot (y \cdot z) = (x \cdot y) \cdot z \tag{5}$$
$$x \cdot y = y \cdot x \tag{6}$$
$$1 \cdot x = x \tag{7}$$
$$x \cdot (y + z) = (x \cdot y) + (x \cdot z) \tag{8}$$

$$\frac{1}{(\frac{1}{x})} = x \tag{9}$$
$$\frac{x \cdot x}{x} = x \tag{10}$$
$$\frac{x}{y} = x \cdot \frac{1}{y} \tag{11}$$

Table 2: Equations for a fracterm calculus of involutive meadows

Although logics of partial functions abound, they are non-trivial technically and require a high level of experience with mathematical logic, e.g. [Robinson 1989, Jones and Middelburg 1994].

6.4 Role and nature of the axiomatisation

One purpose of our axiomatisations is to create a formalism that removes a need for 'type-checking' mathematical texts against the informal principles of (say) the Synthetic Calculus, proposed here to define arithmetic. This idea on managing 'type-checking for conformance' is an aspect of our methodological approach and may be applicable more generally (see section 7.1).

In fact, any attempt at type checking arithmetical texts, even in a narrowly designed language fragment, proves to be very complicated – certainly theoretically undecidable and pragmatically very challenging. We have studied this is some detail in the build-up to the current state of our theory in [Bergstra and Tucker 2023c]. Partiality once present is not easy to control!

The axiomatisation of common meadows given here is polished and aligns well with the style of classical axiomatisations of rings, fields etc. Its status is confirmed by its emerging mathematical theory, e.g., in terms of the equational specification of partiality [Bergstra and Tucker 2022a] and the completeness theorem for equational reasoning [Bergstra and Tucker 2022b]. However, it is worth noting that it is the result of over 15 years of technical investigations, starting with our algebraic specifications of the rational numbers in [Bergstra and Tucker 2007], which used the alternate $0^{-1} = 0$! The theoretical journey through the different aspects of arithmetic – algebra, its roles in computing, textual analysis, its many manifestations in school teaching – has been long and surprising.

7 Concluding remarks

As we have shown, SFTC begins with a broad and liberal consensus identified and mapped by the naive fracterm calculus NFTC in [Bergstra and Tucker 2023]. The conclusions of that paper are relevant here, e.g., about the need to analyse conventions, reasoning patterns, and logics in relation to abstract descriptions of computer arithmetics and, more generally, in requirements analysis and program testing.

As this work demonstrates, bridging the gap between informal descriptions and formal modelling is not easy, even in very technical contexts like arithmetic; this paper is a case study about that matter. In fact, we believe the four stage analysis given for arithmetic in subsection 1.1 has wider application, indeed wherever there is a serious need for formal systems to bear the weight of informal and variable practices.

7.1 The general methodology

This work on arithmetic exemplifies a new methodology for analysing a ‘collection of ideas in circulation’ that could benefit from formal analysis. As pointed out in section 1.1, there are four distinct stages, which we will now describe more abstractly; these are:

1. *Raw Material.* Collect the ideas in circulation that are used in practice – ideas ‘in the wild’.

2. *Naive Categorisation.* Formulate a description of what seems to be a consensus on what ideas and conventions for their use are (i) agreed, (ii) disputed or (iii) remain ambiguous in peoples’ practices.

3. *Synthetic Postulation.* Refine the informal description of the Naive Categorisation to resolve ambiguities and arbitrate on some disputed ideas and conventions and make it a collection of ideas fit for systematic logical reasoning, at least informally.

4. *Formal Codification.* Propose formal or semiformal systems or calculi to pin down the ideas based on the informal analysis of Synthetic stage.

There is no shortage of professional practices where there are ‘collections of ideas in circulation’ that could benefit, from at least the first three stages: the collection itself constitutes the Raw Material; their discovery and cataloging constitutes their Naive Categorisation; and their filtering, confirmation and rejection constitutes their Synthetic Postulation. As to their Formal Codification, this may be a system of rules, conventions, laws, assumptions that are fixed, no longer negotiated (at least for a time), and expected to be followed. In some aspects of life this Formal Codification is a jurisdiction, in others it is a code of best practice, in still others it is an axiomatisation.

In general, the gaps between a pair of stages can vary. The gaps between Raw versus Naive and Synthetic versus Formal are easily understood and will be clearly distinct.

However, the gap between Naive and Synthetic can seem quite small, depending on the nature, volume and variety of the Naive options that are possible and subsequently become Synthetic decisions. This smaller gap is understandable, but it is not a weakness in our theory. The synthetic stage introduces both technical objectives and a lot of technical decisions that we expect the formalisation to accommodate or at least address. In this paper we are dealing with arithmetic, which is familiar and a small and well-explored intellectual space. In every case, in writing about the synthetic stage one has to repeat some of the naive stage for the comparison of options and explanation of decisions; this also makes the gap seem smaller.

7.2 Connection to pedagogy and teaching

The synthetic fracterm calculus (SFTC) we offer here captures a conceptually cautious and precise informal view of elementary arithmetic involving addition, multiplication, subtraction, and division. Such reflections are largely absent from educational texts on teaching arithmetic, e.g., [Kieren 1976]. In contrast with a synthetic view on elementary arithmetic, several formalisations can be developed which may, or many not, be equivalent. As an analytical tool, one objective of formalisation is to detect, make explicit and settle technically subtle points and relations; this process leads to a ramification of options and, so, to a plurality of formal fracterm calculi. We mentioned three formal fracterm calculi here with semantics that are currently recognisable in the schoolroom.

In particular, our work in this paper and its predecessor has uncovered and focussed on some obvious and subtle problems that are visible in teaching rational number arithmetic. By drawing attention to these notions in the Naive calculus and their transition in the Synthetic calculus we are merely preparing the way for pedagogical thinking about practical arithmetic. We hope that SFTC can be used as a point of departure from which to analyse old, and design new, educational material about rational arithmetic, improving the accounts of ambiguous and difficult points, especially for the teacher.

As to the formulation of possible 'experiments' in the classroom stimulated by our ideas, here we have nothing to say at this stage. We believe that the existence of the naive, synthetic and formal axiomatisations are theoretical components that are essential constituents of a foundation for thinking about such pedagogical investigations of arithmetic teaching. Indeed, working on such 'experiments' in advance of a full theoretical analysis is likely to be unproductive. There is even a further stage in which philosophical aspects of this whole exercise on arithmetic are considered in the light of the philosophy of mathematical education.

However, if it comes to teaching elementary arithmetic, we expect that material based on a fracterm calculus for common meadows (i.e., Table 1) will prove to be the most appropriate.

References

[Anderson and Bergstra 2020] Anderson, J. A., Bergstra, J. A.: "Review of Suppes 1957 proposals for division by zero"; Transmathematica. ISSN 2632-9212 (published 06-08-2021), https://doi.org/10.36285/tm.53, (2021).

[Anderson et al. 2007] Anderson, J.A., Völker, N., Adams, A. A.: "Perspecx Machine VIII, axioms of transreal arithmetic"; Vision Geometry XV, J. Latecki, D. M. Mount and A. Y. Wu (eds.), 649902, (2007).

[Andreka et al. 1988] Andreka, H., Craig, W., Nemeti, I.: "A system of logic for partial functions under existence-dependent Kleene equality"; Journal of Symbolic Logic, 53 (3), (1988), 834-839.

[Bergstra 2019] Bergstra, J.A.: "Adams Conditioning and Likelihood Ratio Transfer Mediated Inference"; Scientific Annals of Computing, XXIX (1), 2019, 1Ð58, https://doi.org/10.7561/SACS.2019.1.1

[Bergstra 2019b] Bergstra, J.A.: "Division by zero, a survey of options"; Transmathematica, ISSN 2632-9212, (published 2019-06-25), https://doi.org/10.36285/tm.v0i0.17, (2019).

[Bergstra 2020] Bergstra, J.A.: "Arithmetical datatypes, fracterms, and the fraction definition problem"; Transmathematica, ISSN 2632-9212, (published 2020-04-30), https://doi.org/10.36285/tm.33, (2020).

[Bergstra et al. 2013] Bergstra, J.A., Bethke, I., Ponse, A.: "Cancellation meadows: a generic basis theorem and some applications"; The Computer Journal, 56 (1), (2013), 3–14. Also arxiv.org/abs/0803.3969.

[Bergstra et al. 1995] Bergstra, J.A., Bethke, I., Rodenburg, P.H.: "A propositional logic with 4 values: true, false, divergent and meaningless"; Journal of Applied Non-Classical Logics, 5 (2), (1995), 199–217.

[Bergstra and van de Pol 2011] Bergstra, J.A., van de Pol, J.: "A calculus for four-valued sequential logic"; Theoretical Computer Science. 412, (28), (2011), 3122–3128.

[Bergstra et al. 2009] Bergstra, J.A., Hirshfeld, Y., Tucker, J.V.: "Meadows and the equational specification of division"; Theoretical Computer Science, 410 (12), (2009), 1261–1271.

[Bergstra and Middelburg 2011] Bergstra, J.A., Middelburg, C.A.: "Inversive meadows and divisive meadows"; Journal of Applied Logic, 9 (3), (2011), 203-220. (Also: https://arxiv.org/pdf/0907.0540.pdf.)

[Bergstra and Middelburg 2016] Bergstra, J.A., Middelburg, C.A.: "Transformation of fractions into simple fractions in divisive meadows"; Journal of Applied Logic, 16, (2015), 92–110. (Also: https://arxiv.org/abs/1510.06233)

[Bergstra and Ponse 2011] Bergstra, J.A., Ponse, A.: "Proposition Algebra"; ACM Transactions on Computational Logic, 12 (3), (2011), 1–36. https://doi.org/10.1145/1929954.1929958.

[Bergstra and Ponse 2021] Bergstra, J.A., Ponse, A.: "Division by zero in common meadows"; In R. de Nicola and R. Hennicker (editors), "Software, Services, and Systems (Wirsing Festschrift)", Lecture Notes in Computer Science 8950, Springer, (2015), 46–61. Also in improved form (2021) at: `arXiv:1406.6878v4 [math.RA](2021)`.

[Bergstra and Ponse 2016] Bergstra, J.A., Ponse, A.: "Fracpairs and fractions over a reduced commutative ring"; Indigationes Mathematicae, 27, (2016), 727–748. http://dx.doi.org/10.1016/j.indag.2016.01.007. (Also: https://arxiv.org/abs/1411.4410).

[Bergstra and Ponse 2020] Bergstra, J.A., Ponse, A.: "Arithmetical datatypes with true fractions"; Acta Informatica, 57 (3-5), 385–402. https://doi.org/10.1007/s00236-019-00352-8, (2020).

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

[Bergstra and Tucker 2020] Bergstra, J.A., Tucker, J.V.: "The transrational numbers as an abstract data type"; Transmathematica, ISSN 2632-9212, (published 2020-12-16), https://doi.org/10.36285/tm.47, (2020).

[Bergstra and Tucker 2022a] Bergstra, J.A., Tucker, J.V.: "Partial Arithmetical data types of rational numbers and their equational specification"; Journal of Logical and Algebraic Methods in Programming, 128, (2022), 100797. https://doi.org/10.1016/j.jlamp.2022.100797

[Bergstra and Tucker 2022b] Bergstra, J.A., Tucker, J.V.: "A complete finite equational axiomatisation of the fracterm calculus for common meadows"; https://arxiv.org/abs/2307.04270.

[Bergstra and Tucker 2023] Bergstra, J.A., Tucker, J.V.: "Naive Fracterm Calculus"; J. Universal Computer Science, 29 (9), (2023), 961-987.

[Bergstra and Tucker 2023b] Bergstra, J.A., Tucker, J.V.: "On the axioms of common meadows: Fracterm calculus, flattening and incompleteness"; The Computer Journal, 66 (7), 2023, 1565–1572. https://doi.org/10.1093/comjnl/bxac026.

[Bergstra and Tucker 2023c] Bergstra, J.A., Tucker, J.V.: ”On legality of texts involving fracterms: an exercise in philosophical arithmetic”; Submitted, 2023.

[Bethke and Rodenburg 2010] Bethke, I and Rodenburg, P.H.: “The initial meadows”; Journal of Symbolic Logic, 75 (3), (2010), 888-895.

[Bethke et al. 2015] Bethke, I., Rodenburg, P.H., Sevenster, A.: “The structure of finite meadows”; Journal of Logical and Algebraic Methods in Programming, 84 (2), (2015), 276–282.

[Bidoit and Mosses 2004] Bidoit M. and Mosses, P. D.: ”Casl User Manual - Introduction to Using the Common Algebraic Specification Language”; Lecture Notes in Computer Science 2900, Springer, 2004. https://doi.org/10.1007/b11968

[Carlström 2004] Carlström, J.: “Wheels – on division by zero”; Math. Structures in Computer Science, 14 (1), (2004), 143-184.

[Fandino Pinilla 2007] Fandino Pinilla, M.I.F.: “Fractions: conceptual and didactic aspects”; Acta Didactica Universitatis Comenianae, 7, (2007), 82-115.

[Halmos 1960] Halmos, P.: Naive Set Theory; D. Van Nostrand Company, 1960.

[Jones and Middelburg 1994] Jones, C.B., Middelburg, C.A.: “A typed logic of partial functions, reconstructed classically”; Acta Informatica, 31 (1994), 399-430.

[Kieren 1976] Kieren, T.E.: “On the mathematical, cognitive, and instructional foundations of rational numbers”; In R.A. Lesh and D.A. Bradbart (editors), Number and Measurement. Papers from a Research Workshop. ERIC, Columbus Ohio, (1976), 101–144, (available at http://files.eric.ed.gov/fulltext/ED120027.pdf#page=108, accessed 1 September, 2023).

[Michikawi et al. 2016] Michiwaki, H., Saitoh, S., Yamada, N.: “Reality of the division by zero $z/0 = 0$”; International Journal of Applied Physics and Mathematics, doi: 10.17706/ijapm.2016.6.1.1-8 (2016).

[Morgan 2023] Morgan, J, “Resourceaholic. Ideas and resources for teaching secondary school mathematics: Online Historical Maths Textbooks”. https://www.resourceaholic.com/p/digitised-antique-maths-textbooks.html.

[Ono 1983] Ono, H.: “Equational theories and universal theories of fields”; Journal of the Mathematical Society of Japan, 35 (2), (1983), 289–306.

[Okumura 2018] Okumura H.: “Is it really impossible to divide by zero?”; Biostatistics and Biometrics Open Access Journal, 7 (1), (2018) 555703. DOI: 10.19080/BBOJ.2018.07.555703, (2018)

[Ponse and Staudt 2018] Ponse, A., Staudt, D.J.C.: “An independent axiomatisation for free short-circuit logic”; Journal of Applied Non-Classical Logics, 28 (1), (2018), 35–71.

[Robinson 1989] Robinson, A.: “Equational logic of partial functions under Kleene equality: a complete and an incomplete set of rules”; Journal of Symbolic Logic, 54 (2), (1989), 354–362.

[Rodenburg 2001] Rodenburg, P.H: “A complete system of four-valued logic”; Journal of Applied Non-classical Logics, 11 (3/4), (2001), 367–389.

[Setzer 1997] Setzer, A.: “Wheels (draft)”; http://www.cs.swan.ac.uk/csetzer/articles/wheel.pdf, (1997).

[Suppes1975] Suppes, P.: “Introduction to Logic”; Van Nostrand Reinhold, 1957.

[van Engen 1960] van Engen, H.: “Rate pairs, fractions, and rational numbers”; The Arithmetic Teacher, 7 (8), (1960), 389–399.