A Complete Finite Axiomatisation of the Equational Theory of Common Meadows

Jan A Bergstra[1] and John V Tucker[2]

[1] Informatics Institute, University of Amsterdam, Science Park 900, 1098 XH, Amsterdam, The Netherlands
j.a.bergstra@uva.nl
[2] Department of Computer Science, Swansea University, Bay Campus, Fabian Way, Swansea, SA1 8EN, United Kingdom
j.v.tucker@swansea.ac.uk

Abstract. We analyse abstract data types that model numerical structures with a concept of error. Specifically, we focus on arithmetic data types that contain an error value $\perp$ whose main purpose is to always return a value for division. To rings and fields, we add a division operator x/y and study a class of algebras called *common meadows* wherein $x/0 = \perp$. The set of equations true in all common meadows is named the *equational theory of common meadows*. We give a finite equational axiomatisation of the equational theory of common meadows and prove that it is complete and that the equational theory is decidable.

Keywords: arithmetical data type, division by zero, error flag, common meadow, fracterm, fracterm calculus.

1 Introduction

Arithmetical structures have deep mathematical theories exploring their abstract axiomatisations, concrete representations, comparisons by homomorphisms, use in constructions, methods of equation solving, etc. For example, the naturals form commutative semirings, the integers form commutative rings, and the rationals, reals and complex numbers form fields. However, for computing, their classical algebraic theories have some shortcomings. Computing with arithmetical structures requires us to make abstract data types with extra algebraic properties that arise from the semantics of algorithms and programs. In practical computation, the application of an operator must always return a value, i.e., it must be a total operator. For this reason arithmetical structures in computing can have various special elements that indicate special behaviour; the most obvious examples are error values, such as a pocket calculator displays when trying to compute $1/0$ or when having an overflow. Floating point arithmetics employ several more values, such as infinities $+\infty, -\infty$ and 'not a number' NaN. Surprisingly, not much is known about the algebraic theories of these augmented structures whose semantical features have been deemed practically essential for arithmetical abstract data types. What has been known, at least since von Neumann and Goldstine's 1947 analysis of numerics, is that computer arithmetics do not satisfy the beautiful axioms of classical algebra [35,56].

1.1 Common meadows

In [14], we began to investigate semantic aspects of computer arithmetic using the theory of abstract data types. Using the equational methods characteristic of the theory, we have studied several semantic options for undefined operators and overflows, often focussing on data types of rational numbers (we sketch some of this programme later, in section 6.2).

In this paper, we consider the class of arithmetical data types called *common meadows*, which have the general form

$$(F \cup \{\bot\} \mid 0, 1, \bot, x + y, -x, x \cdot y, x/y)$$

where F is a field and $\bot$ is an element that behaves like an error value. Following [14,8], we use the term *meadow* for any field equipped with an explicit operator for division, or inverse. The idea of a common meadow was introduced in [11]. The class of all common meadows is denoted CM.

Common meadows are built from fields by adding error and division, as follows. Given any field F, we extend its domain with a new element $\bot$ which is *absorptive*, which means for all $x \in F$,

$$x + \bot = \bot, x \cdot \bot = \bot, \text{and} - \bot = \bot.$$

This gives us the enlarged field-like structure $\mathsf{Enl}_\bot(F)$, using the general methods of [18]. The addition of $\bot$ disturbs the classical algebra of fields as standard properties can fail, e.g.,

$$x - x = 0 \text{ fails because } \bot - \bot = \bot \text{ and } x \cdot 0 = 0 \text{ fails because } \bot \cdot 0 = \bot.$$

We will explore the effect of $\bot$ and show that, surprisingly, many familiar laws can be preserved or rescued.

With $\bot$ installed, we can extend $\mathsf{Enl}_\bot(F)$ with a total division function $\frac{x}{y}$, also written x/y, and defined by:

$\frac{x}{y} = \bot$ if $y = 0$, $y = \bot$ or $x = \bot$; otherwise,

$\frac{x}{y} = x \cdot y'$ where $y' \in F$ is the unique element for which $y \cdot y' = 1$ in F.

This algebra is denoted $\mathsf{Enl}_\bot(F(\_/\_))$ and is a common meadow.

With these constructions introduced, we can now turn to the main theorem of the paper, for which we need to be very precise about the syntax of rings, fields and common meadows. The syntax is determined by choosing signatures that contain names for the constants and operations. We need several: Σ_r for rings and fields; $\Sigma_{r,\bot}$ for rings and fields with $\bot$; Σ_m for meadows; and Σ_{cm} for common meadows. We will use terms and equations over these signatures.

1.2 Equational theory of common meadows

The importance of the field of rational numbers for computing influences our use of rings and fields in developing data types. Earlier, we have sought finite axiomatisations to capture the algebraic laws of common meadows, taking the axioms of rings and fields as an inspiration and guide. This has led, in [17], to a particular equational axiomatisation $E_{\mathsf{ftc-cm}}$ that has a clear relation with rings and is the main object of study in this paper.

In addition to focussing on division as a total function, we highlight the idea of a fraction – the primary representation of rationals in practice – adapting it to the abstract setting of meadows. Although fractions are not well-defined notions, the idea can be made perfectly precise using the syntax of the signature containing division.

Definition 1. *A fracterm is a term over the meadow signature Σ_m whose leading function symbol is division. Since the equations of $E_{\mathsf{ftc-cm}}$ highlight fracterms, we call $E_{\mathsf{ftc-cm}}$, equipped with the standard rules for equational deduction, a fracterm calculus.*[3]

Definition 2. *The* equational theory of common meadows *is the set*

$$Eqn(\mathsf{CM}) = \{e \mid \forall A \in \mathsf{CM}. A \models e\}$$

of all equations over Σ_{cm} that are true in all common meadows.

The objective of the paper is to develop enough theory to prove the following new result (Theorem 4 below).

Theorem. *The finite equational axiomatisation $E_{\mathsf{ftc-cm}}$, equipped with equational logic, is sound for the class* CM *of all common meadows, and complete for the equational theory $Eqn(\mathsf{CM})$ for common meadows. Thus, for any equation e over Σ_{cm},*

$$E_{\mathsf{ftc-cm}} \vdash e \textit{ if, and only if, } e \in Eqn(\mathsf{CM}).$$

Corollary. *The equational theory for common meadows is algorithmically decidable.*

So, in the language of logic, the equational theory of common meadows is finitely based and decidable.

The class of *all* fields is classically definable by finitely many first order axioms; but it is not definable by any set of equations or conditional equations as they do not form a variety in the sense of Birkhoff's Theorem, or a quasivariety in the sense of Mal'tsev's Theorem (as they are not closed under products) [38,40]. The same is true of the class CM of all common meadows. The fact about fields is the classic illustration of consequences of Birkhoff's remarkable foundational analysis of universal algebras of 1935 [21].

[3] Fracterms were introduced in [12], and a full motivation for the use of this syntax and terminology is given in [5].

Equations, and conditional equations, are the preferred forms of axioms for data types, especially as they have good term rewriting properties [13]; they are a basic component for specification and verification tools. Seeking equational specifications of arithmetical data types is a technical programme for which completeness is something of an aspiration. Common meadows have emerged as a mathematically attractive and tractable data type semantics for specifying and reasoning about computer arithmetic. Our theorem improves on earlier axiomatisations and on a partial completeness result for common meadows given in [11], based on fields with characteristic 0.

Complementing our theorem here is the fact, proved in [17], that our axiomatisation $E_{\mathsf{ftc-cm}}$ does *not* prove all conditional equations even for characteristic 0:

Question. Does the conditional equational theory of common meadows have a sound and complete finite conditional equation axiomatization?

1.3 Structure of the paper

We begin with preliminaries. First, we recall basic ideas about abstract data types in section 2 that we will use and, indeed, situates our research programme. Secondly, in section 3, we consider concepts to do with rings, fields and common meadows that we use and are the foundation of our algebraic approach to computer arithmetics. Polynomials play a central role in all arithmetical structures and so transitions between standard polynomials and syntactic polynomials for rings, fields and common meadows are established in section 4. In section 5 we use the ideas and results we have accumulated to prove the theorems. Finally, in section 6, we explicate and situate the results in logic, reflect on our programme, and discuss some open problems that arise naturally.

The results of this paper are relate to abstract data type theory, computer arithmetic, algebra and logic. We have tried to make the paper sufficiently self-contained to serve the needs of these audiences. Our preliminary material is designed to recall key ideas and results, and to settle notation, and include many pointers to the literature for further explanations. We do assume that the reader has some knowledge of equational specifications of data types, rings and fields, and first order logics.

We thank two referees for their questions, comments and suggestions, which have enabled us to improve the paper.

2 Preliminaries on data types

The theory of abstract data types starts from four basic concepts as follows. An implementation of a data type is modelled by a many-sorted algebra A of signature Σ. A signature Σ is an interface to some (model of an) implementation of the data type, and the constants and operations declared in Σ provide the only means of access to the data for the programmer. Axiomatisations of the operations in a signature define a range of implementations and provide the only means for the programmer to reason about the data. Two implementations of an interface are equivalent if, and only if, their algebraic models are isomorphic. The theory of arithmetic data types we are developing here is shaped by these and the following following general concepts.

2.1 Terms and equations

That signatures model interfaces establishes an essential role for the syntax of terms and equations in the theory abstract data types.

Let Σ be any signature. Let X be any countable set of variables. Let $T(\Sigma)$ and $T(\Sigma, X)$ be the algebras of all closed or ground terms over Σ, and open terms with variables in X, respectively. Given a Σ-algebra A, and a valuation σ for variables in a term $t \in T(\Sigma, X)$, the result of evaluating t in A using σ is denoted $[\![t]\!]_\sigma$.

Definition 3. *An* equation *over the set X of variables is a formula of the form*

$$e \equiv t(x_1, \ldots, x_k) = t'(x_1, \ldots, x_k)$$

where $t(x_1, \ldots, x_k), t'(x_1, \ldots, x_k)$ are terms over Σ with variables from the list $x_1, \ldots, x_k \in X$; note the terms t and t' need not have the same variables. Let $Eqn(\Sigma, X)$ to be the set of all equations over Σ with variables taken from X.

Definition 4. *An equation $e \equiv t = t' \in Eqn(\Sigma, X)$ is* valid *in the Σ algebra A, written $A \models e$, if for all valuations σ of variables of e, $[\![t]\!]_\sigma = [\![t']\!]_\sigma$. The equation e is* valid *in a class K of Σ-algebras, written $\mathsf{K} \models e$, if it is valid in every algebra in K.*

Definition 5. *Let $E \subset Eqn(\Sigma, X)$ be a set of equations over Σ. Then E together with the standard rules of equational deduction forms an* equational calculus*. We write $E \vdash e$ if equation $e \in Eqn(\Sigma, X)$ can be deduced from E.*

The following is a basic fact about reasoning:

Lemma 1. *Let E be a computably enumerable set of equations. Then $\{e | E \vdash e\}$ is computably enumerable.*

Definition 6. *Let K be a class of Σ-algebras. An set E of equations is* sound *w.r.t. equational logic for K if for all equations e, if $E \vdash e$ then $\mathsf{K} \models e$. Conversely, the set E of equations is* complete *w.r.t. equational logic for K if for all equations e, if $\mathsf{K} \models e$ then $E \vdash e$.*

The search for an axiomatisation is a method for discovering the essential properties of some class K of structures of interest. In trying to axiomatise a given class K of structures by a set of equations E, soundness is a necessary property, of course: the equations and formulae logically derivable from them must be true in *all* the models of the axioms in E. However, in the class of *all* models of the axioms E 'non-standard' structures appear that are very different from the structures of K that motivated E. The special case of the class K being an isomorphism type, i.e., K consisting of all structures that are isomorphic to a single structure, is central in computing and is at the heart of abstract data type theory.[4] For any given class K of structures completeness is more complicated and, in fact, can be rare though not unknown. We return to this important topic in section 6.

Definition 7. *Let* K *be a class of* Σ*-algebras. The set*

$$Eqn(\mathsf{K}) = \{e \mid \forall A \in \mathsf{K}.A \models e\}$$

of equations is called the equational theory *of* K*.*

2.2 Data types and their enlargements by $\perp$

The properties of interest to abstract data types are isomorphism invariants – typical examples are properties that are definable by first order formulae and forms of computability. This means that if a property is true of *any* data type A, and is an isomorphism invariant, then the property will be true of its abstract data type. For more of the general theory of abstract data types see [29,30,58,40].

Our algebras will be single-sorted and have a non-empty carrier so we will use a simple notation for data types. For instance,

$$(A \mid c_1, \ldots, c_k, f_1, \ldots, f_l)$$

denotes a data type with domain A and constants c_1, ...,c_k from A, and functions f_1,..,f_k, where it is assumed that arities for the functions on A are known from the context.

Definition 8. *An algebra A is* total algebra *if all its operations are total functions. An algebra A is* partial algebra *if one or more of its operations are partial functions.*

Definition 9. *A Σ-algebra A is* Σ-minimal *if it is generated by the constants and operations named in its signature Σ. A* data type *is a Σ-minimal algebra. An* abstract data type *is an isomorphism class of a data type.*

Definition 10. *An algebra can be* expanded *by adding new constants and operations to its signature. An algebra can be extended by adding new elements to its carriers. Combining expansions and extensions in some order constitutes what we call an enlargement of an algebra.*

Consider the following general method of enlarging an algebra with $\perp$.

[4] One consequence of Skolem is: no first-order theory with an infinite model can have a unique model up to isomorphism.

Definition 11. *Consider the algebra*

$$(A \mid c_1, \ldots, c_k, f_1, \ldots, f_l)$$

of signature Σ. Suppose $\perp \notin A$ and let

$$Enl_{\perp}(A) = (A \cup \{\perp\} \mid c_1, \ldots, c_k, \perp, f_1, \ldots, f_l)$$

wherein $\perp$ is

(i) absortive*, i.e., if $\perp$ is an argument to an operation f then the result is $\perp$; and*

(ii) totalising*, i.e., if any operation f is undefined in A then it returns $\perp$ in $Enl_{\perp}(A)$.*

Let $\Sigma_{\perp} = \Sigma \cup \{\perp\}$ be the signature of $Enl_{\perp}(A)$.

If the algebra A is total then f returns $\perp$ if, and only if, one of its arguments is $\perp$.

We can adapt some equational axioms true of A to accommodate $\perp$ by using this idea:

Definition 12. *An equation $t = t'$ is a* balanced equation *if the terms t and t' have the same variables.*

Their key property is this:

Lemma 2. *Let A be a Σ algebra and let $t = t'$ be a balanced equation. Then,*

$$A \models t = t' \text{ if, and only if, } Enl_{\perp}(A) \models t = t'.$$

3 Preliminaries on arithmetic structures

In the arguments that follow, we will move between the algebra of rings, fields and common meadows.

3.1 Rings and fields and common meadows

We start from the theory of commutative rings and fields.

Definition 13. *A* commutative ring with 1 *is an algebra R of the form*

$$(R \mid 0, 1, x + y, -x, x \cdot y)$$

satisfying the axioms of Table 1.

Definition 14. *A* field *is a commutative ring F with 1 in which $0 \neq 1$ and for all $x \in F$,*

$$x \neq 0 \text{ implies } \exists y[x \cdot y = 1].$$

Let Σ_r be a signature for rings and fields. All our rings will be commutative with 1. Note rings and fields have the same three operations.

Let $\mathbb{Z}$ be a ring of integers and let $\mathbb{Q}$ be a field of rational numbers containing the subring $\mathbb{Z}$.

Definition 15. *In a ring R, for each x, if there is a y such that $x \cdot y = 1$ then x is called an* invertible element *and y is called the* inverse *of x.*

In all rings, the additive inverse 0 is not invertible as we can derive $0.x = 0$ from Table 1.

Definition 16. *A* field *is a ring in which all elements are invertible, except* 0*.*

In many rings, the inverse y of an invertible element x is unique, and so an explicit operator, with a familiar notation $^{-1}$, can be introduced for calculating $y = x^{-1}$. The operator x^{-1} is partial as it is only defined for invertible elements. Thus, a derived division operator $x/y = x \cdot y^{-1}$ is also partial for $x = 0$ on a field.

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

Table 1. E_{cr}: equational axioms for commutative rings with 1

It is perhaps worth noting that the definition of a field as a special type of ring – thus having only ring operations and no inverse or division – was well established from the early days of abstract algebra. The axiomatic approach was to focus on equation solving such as $ax = b$ in rings [57], and on the invertible elements in rings [27,36,22]. These approaches are also to be found in universal algebra and model theory [38,33]. The inverse operator was not used for defining fields, not least because of partiality, though there are examples in some student textbooks and lecture notes, e.g., [50]. As the algebraic consequences of the act of introducing division into a ring is an object of our theory of meadows, our definition of field follows strictly the classical tradition.

Definition 17. *By applying the enlargement of Definition 11, we add $\perp$ to a ring R to build the algebra*

$$Enl_{\perp}(R) = (A \cup \{\perp\} \mid 0, 1, \perp, x + y, -x, x \cdot y)$$

with signature $\Sigma_{r,\perp}$. The same construction applied to a field F yields $Enl_{\perp}(F)$.

The point of adding $\perp$ is to manage the partiality of division.

3.2 Equation solving and algebraically closed fields

We will call upon some basic theory of rings and fields in what follows. In particular, the classical theory of polynomials plays an important role in our arguments. There are many classic [57,27,36,22] and contemporary textbooks on rings and fields to which reference can be made for what we need. Here we recall a few important notions to do with the algebra of solving polynomial equations.

Definition 18. *Let $F[X]$ be the set of polynomials with variable X. An element $a \in F$ is a* root *of a polynomial $p \in F[X]$ if $p(a) = 0$ in F.*

Roots are key to the factorisation of polynomials: if a is a root of p then p is divisible by $(X - a)$.

Definition 19. *A polynomial $p \in F[X]$ is* irreducible *over F if it cannot be factored into the product of two non-constant polynomials with coefficients in F.*

For many fields not every polynomial in has a root. Most notably, for $F = \mathbb{R}$, a field of real numbers, $p(X) = X^2 + 1$ does not have a root in $\mathbb{R}$. For this situation the extension to a field of complex numbers $\mathbb{C}$ was created wherein every polynomial over $\mathbb{C}$ – and thus over $\mathbb{R}$ – had a solution and the number of solutions corresponded with the degree of the polynomial – a result finally proved by Carl Friedrich Gauss in 1799 and celebrated as the 'Fundamental Theorem of Algebra'.

Basic field theory generalises the solution of polynomial equations.

Definition 20. *A field F is* algebraically closed *if every polynomial $p \in F[X]$ has a root in F.*

Theorem 1. *For each field F, there exists a field K containing F that is algebraically closed. Furthermore, there is a smallest such field $\overline{F}$, called the* algebraically closure *of F, that is unique as an extension of F up to isomorphism.*

Definition 21. *A field F is* prime *if it contains no subfields.*

The finite prime fields F_p are isomorphic to $\mathbb{Z}_p$, the modulo p arithmetics for p a prime; the infinite prime fields are isomorphic to the rationals $\mathbb{Q}$. Every field contains a subfield that is prime and so isomorphic to either $\mathbb{Z}_p$ or $\mathbb{Q}$.

In sections 4 and 5, we will use the algebraic closures $\overline{F_p}$ and $\overline{\mathbb{Q}}$ of prime fields.

3.3 Meadows and common meadows

To fields we add division operator to make a meadow.

Definition 22. *A* meadow *is a partial algebra $F(\_/\_)$ obtained as an expansion of a field with a division function $\_/\_$ that works as usual on non-zero elements of the domain of F. Let $\Sigma_m = \Sigma_r \cup \{\_/\_\}$.*

To totalise division, we add $\perp$ to a meadow $F(\_/\_)$ by applying the enlargement of Definition 11:

Definition 23. *A* common meadow *is a total algebra*

$$Enl_{\perp}(F(\_/\_)) = (F \cup \{\perp\} \mid 0, 1, \perp, x + y, -x, x \cdot y, x/y)$$

with signature $\Sigma_{cm} = \Sigma_{m,\perp}$.

Thus, we have a field F equipped with a division function $\_/\_$ that has been made total by having $x/0 = \perp$ for all x, including $\perp$.[5]

Recall that to qualify as a data type, an algebra must be minimal, i.e., generated by its constants and operations (Definition 9). Now, if F_p is a finite prime field (isomorphic to modulo p arithmetic on $\{0, 1, \dots, p\}$, for p a prime number) then $\mathsf{Enl}_{\perp}(F_p)$ is minimal. For all other fields F – in particular, the rationals – the algebra is non-minimal and is not a data type for that reason. Division is needed to make the classical field of rational numbers a data type:

Lemma 3. *The common meadow $Enl_{\perp}(\mathbb{Q}(\_/\_))$ of rationals is Σ_{cm}-minimal and hence qualifies as a data type.*

Proof. The ring operations of $+, -, \cdot$ applied to constants $0, 1$ generate the integers only. But with the operation of division $\_/\_$ all rational numbers can be constructed.

Recalling an observation made in [11], we summarise the constuction:

Proposition 1. *Every field F can be enlarged to a common meadow $Enl_{\perp}(F(\_/\_))$ that is unique with respect to isomorphisms that fix the field F.*

An algebra is *computable* if its carrier set is decidable, equality between elements of the carrier set is decidable, and the operations of the algebra are computable.

Proposition 2. *If F is a computable field then $Enl_{\perp}(F(\_/\_))$ is a computable common meadow.*

Proof. It is easy to see that the extension of F by $\perp$ is computable. Division is partial on F, but its set $\{(x, 0) | x \in F\}$ of undefined arguments is computable, for which the value $\perp$ for divisions can be computed. See, e.g., [51] for methods to express this argument about fields in detail.

[5] Equivalent designs for meadows and common meadows can be based on inverse as a primitive, an approach that was taken in [11].

Applying the definitions of equations in section 2.1 we have:

Definition 24. *The* equational theory of common meadows *is the set*

$$Eqn(\mathsf{CM}) = \{e \in Eqn(\Sigma_{cm}) \mid \forall A \in \mathsf{CM}. A \models e\}$$

of all equations made of Σ_{cm}-terms that are true in all common meadows.

3.4 Polynomial sumterms

For the next steps in preparing for the proof, we need some syntactic theory of polynomials adapted to the presence of $\perp$ in rings and fields and, later, to working with division in common meadows.

Definition 25. *A* sumterm *is a Σ_r term s with $\_ + \_$ as its leading function symbol.*

A pure product term *is a Σ_r term s containing only multiplications $\_ \cdot \_$.*

A flat sumterm *is an arbitrarily long sum $s_1 + \ldots + s_k$ of pure product terms; note that in the presence of associativity we need not employ brackets.*

Let $Eqn(\Sigma_r)$ denote the set of all equations made from terms over Σ_r. Now since

$$\Sigma_r \subset \Sigma_{r,\perp} \subset \Sigma_{cm}$$

the ring terms and equations over Σ_r are destined to play a special role in the theory of common meadows: they are the simple terms and equations over Σ_{cm} that do not involve $\perp$ or division.

Let $SumEqn(\Sigma_r) \subset Eqn(\Sigma_r)$ be the set of all equations whose terms are sumterms.

Definition 26. *The* sumterm equational theory of common meadows *is the set*

$$SumEqn(\mathsf{CM}) = \{e \in SumEqn(\Sigma_r) \mid \forall A \in \mathsf{CM}. A \models e\}$$

of all sumterm equations true in all common meadows.

3.5 Equational specifications with $\perp$

Consider the set $E_{\mathsf{wcr},\perp}$ of equational axioms over $\Sigma_{r,\perp}$ in Table 2.

Notice these equations are close to the equational axioms of commutative rings. The eight equations for commutative rings in Table 2 that are intact are balanced equations (Lemma 2). The two axioms (4) and (10) are adjusted to the presence of $\perp$. For example, the unbalanced equation $x + (-x) = 0$ is replaced by the balanced $x + (-x) = 0 \cdot x$, which is valid for $x = \perp$. Axiom (11) introduces $\perp$, from which the absorption axioms for $\cdot$ and $-$ can be derived from $E_{\mathsf{wcr},\perp}$. We call these axioms for *weak commutative rings*.

An algebra satisfying the axioms for commutative rings with 1 in Table 1 will also satisfy the axioms 1–10 in Table 2. The converse is not the case as the common meadow $Enl_{\perp}(\mathbb{Q}(\_/\_))$ will be seen to be an example.

$$(x+y)+z = x+(y+z) \tag{9}$$
$$x+y = y+x \tag{10}$$
$$x+0 = x \tag{11}$$
$$x+(-x) = 0\cdot x \tag{12}$$
$$x\cdot(y\cdot z) = (x\cdot y)\cdot z \tag{13}$$
$$x\cdot y = y\cdot x \tag{14}$$
$$1\cdot x = x \tag{15}$$
$$x\cdot(y+z) = (x\cdot y)+(x\cdot z) \tag{16}$$
$$-(-x) = x \tag{17}$$
$$0\cdot(x+y) = 0\cdot(x\cdot y) \tag{18}$$
$$x+\bot = \bot \tag{19}$$

Table 2. $E_{\mathsf{wcr},\bot}$: equational axioms for weak commutative rings with $\bot$

Theorem 2. *The equations $E_{\mathsf{wcr},\bot}$ in Table 2 are a finite axiomsatistion that is complete for the*

(i) sumterm equational theory for rings equipped with $\bot$;
(ii) sumterm equational theory for fields equipped with $\bot$; and
(iii) sumterm equational theory for common meadows.

Proof. The validity of these axioms in all structures of the form $\mathsf{Enl}_\bot(F(\_/\_))$, for a field F, is easy to check by inspection. Hence, the axioms are sound for $SumEqn(\mathsf{CM})$.

By Proposition 2.3 of [17], the equations $E_{\mathsf{wcr},\bot}$ of Table 2 provide a complete axiomatisation of the equational theory of the class of structures obtained as $\mathsf{Enl}_\bot(R)$ for some ring R. It is an immediate corollary of the proof of Proposition 2.3 in [17] that contemplating a smaller class of structures by requiring that R is a field allows the conclusion to be drawn for $\mathsf{Enl}_\bot(F)$: In the final lines of that proof, instead of considering a ring of integers one may use, to the same effect, a field of rationals. Since the sum terms and equations do not involve division, completeness holds $\mathsf{Enl}_\bot(F(\_/\_))$.

In section 5, we build the equations of common meadows by axiomatising division $\_/\_$ on top of this set $E_{\mathsf{wcr},\bot}$.

4 Standard polynomials as syntactic terms over common meadows

In conventional algebra, working with standard polynomials over rings and fields does not involve syntax nor, of course, $\bot$. Here we collect some results on standard polynomials over fields and, in particular, (i) formalise syntactic terms for standard polynomials and (ii) establish a two-way transformation between standard polynomials and their formal syntactic counterparts.

4.1 Properties of standard polynomials and algebraically closed fields

Consider the polynomial rings $\mathbb{Z}[X_1, \ldots, X_n] \subseteq \mathbb{Q}[X_1, \ldots, X_n]$. We need to distinguish and restrict attention to specific types of multivariate polynomials.

Definition 27. *A* coefficient *of a polynomial in $\mathbb{Z}[X_1, \ldots, X_n]$ or $\mathbb{Q}[X_1, \ldots, X_n]$ is any number multiplying some variables in the polynomial.*

Thus, any polynomial containing, say, the term $0 \cdot X_1 \cdot X_2$ will not be considered a polynomial in $\mathbb{Z}[X_1, X_2]$ with non-zero coefficients. Each number $s \in \mathbb{Q}$, including 0, counts as a polynomial with non-zero coefficients.

Definition 28. *A polynomial p in $\mathbb{Z}[X_1, \ldots, X_n]$ is primitive if the greatest common divisor of its coefficients is* 1*.*

Recalling subsection 3.2, let $\overline{\mathbb{Q}}$ be an arbitrary but fixed algebraic closure of the field $\mathbb{Q}$.

Proposition 3. *Suppose p and q are polynomials in $\mathbb{Q}[X_1, \ldots, X_n]$ which take value* 0 *at the same argument vectors in $\overline{\mathbb{Q}}^n$, then p and q have the same irreducible polynomials as factors (up to constant factors in $\mathbb{Q}$), in the ring $\mathbb{Q}[X_1, \ldots, X_n]$.*

Proof. This follows by repeated application of the Nullstellensatz (e.g., [37], Ch. IX, Theorem 1.5) and unique factorization (e.g., [37], Ch. IV, Corollary. 2.4).

Proposition 4. *(Lemma of Gauss.) Consider a polynomial $p \in \mathbb{Z}[X_1, \ldots, X_n]$. Suppose that p is non-zero and has a factorisation $p = r_1 \cdot r_2$ in $\mathbb{Q}[X_1, \ldots, X_n]$. Then for some numbers $c_1, c_2 \in \mathbb{Q}$, $p = c_1 \cdot r_1 \cdot c_2 \cdot r_2$ and the polynomials $c_1 \cdot r_1$ and $c_2 \cdot r_2$ are in $\mathbb{Z}[X_1, \ldots, X_n]$.*

Proposition 5. *Suppose that a non-zero primitive polynomial $p \in \mathbb{Z}[X_1, \ldots, X_n]$ has a factorisation $p = r_1 \cdot \ldots \cdot r_m$ with $r_1, \ldots, r_m$ irreducible polynomials in $\mathbb{Z}[X_1, \ldots, X_n]$. Then the multiset $\{r_1, \ldots, r_m\}$ of polynomials, modulo the sign thereof, is unique.*

Proposition 6. *Suppose α and β are primitive non-zero polynomials in the ring $\mathbb{Z}[X_1, \ldots, X_n]$ with the property that α and β take value* 0 *on the same argument vectors in $\overline{\mathbb{Q}}^n$. Then there are primitive irreducible polynomials $\gamma_1, \ldots, \gamma_m \in \mathbb{Z}[X_1, \ldots, X_n]$ and positive natural numbers $a_1, \ldots, a_n, b_1, \ldots, b_m$ such that in $\mathbb{Z}[X_1, \ldots, X_n]$,*

$$\alpha = \gamma_1^{a_1} \cdot \ldots \cdot \gamma_n^{a_m} \text{ and } \beta = \gamma_1^{b_1} \cdot \ldots \cdot \gamma_n^{b_m}.$$

Proof. By Proposition 3, if in $\overline{\mathbb{Q}}$ it is the case that α and β vanish on the same arguments both have the irreducible factors, say $\gamma_1, \ldots, \gamma_m$ over $\mathbb{Q}[X_1, \ldots, X_n]$. Using Proposition 4, these irreducible polynomials may be chosen in $\mathbb{Z}[X_1, \ldots, X_n]$, and with Proposition 5 one finds that, viewed as a set, said collection of polynomials is unique modulo the sign of each polynomial.

In the proof below only Proposition 6 will be used.

4.2 Polynomial sumterms in the setting of common meadows

The step from the ordinary algebra of rings and fields to working with equational logic in common meadows is a step from informal semantical practice to a formal syntax and semantics. It is not difficult, but it involves some details.

The key syntactic idea is a special sumterm called a *polynomial sumterm* over Σ_r, and hence over our other signatures, which will work like a standard polynomial in conventional algebra.

To replicate in syntax the various standard polynomials, we begin with choosing sets of numerals, which are closed terms for denoting the naturals, integers and rationals. Numerals for natural numbers are: $0, 1, \underline{2}, \underline{3}, \ldots$ where $\underline{2} \equiv 1 + 1, \underline{3} \equiv \underline{2} + 1, \ldots$ In general: $\underline{n+1} \equiv \underline{n} + 1$. (The precise definition of numerals is somewhat arbitrary and other choices are equally useful.)

For integers we will have terms of the form $-\underline{n}$ with $n > 0$. We will use the notation $\underline{n}$ for an arbitrary integer, thus $\underline{0} \equiv 0$, $\underline{1} \equiv 1$ and for positive n, $\underline{-n} \equiv -(\underline{n})$.

For rational numbers, we have terms of the form $\dfrac{\underline{n}}{\underline{m}}$ and $-\dfrac{\underline{n}}{\underline{m}}$ with $n > 0, m > 0$ and $\gcd(n, m) = 1$. In this way, for each $a \in \mathbb{Q}$ we have a unique numeral t_a such that $[\![t_a]\!] = a$ in $\mathbb{Q}$.

We build the polynomial sumterms in stages.

Definition 29. *A* pure monomial *is a non-empty product of variables (understood modulo associativity and commutativity of multiplication).*

A monomial *is a product $c \cdot p$ with c a non-zero numeral for a rational number and p a pure monomial.*

We will assume that pure monomials are written in a uniform manner mentioning the variables in the order inherited from the infinite listing $X_1, X_2, \ldots$ with powers expressed as positive natural numbers (where power 1 is conventionally omitted). Recalling Definition 25 of sumterms:

Definition 30. *A* polynomial sumterm *p over Σ_r is a flat sumterm (Definition 25) for which*

(i) all summands involve pairwise different pure monomials, and
(ii) none of the coefficients is 0.

The idea of polynomial sumterms is that these formalise syntactically the notion of standard polynomials with non-zero coefficients. Moreover, in the case of (i),

$1 \cdot x + 1 \cdot x$ would fail while $(1+1) \cdot x$ is a sumterm. Also, 0 is a polynomial sumterm while $\perp$ is not a polynomial sumterm, as polynomial sumterms are terms over Σ_r:

Definition 31. *A polynomial sumterm p is* non-zero *if it contains a variable or if it contains non-zero constant.*

Clearly, if a non-zero polynomial sumterm contains a variable then it must have at least one pure monomial.

Proposition 7. *Given polynomial sumterms p and q,*

$$Enl_{\perp}(\mathbb{Q}) \models p = q \textit{ if, and only if, } E_{\mathsf{wcr},\perp} \vdash p = q.$$

Proof. This is an immediate corollary of the proof of Theorem 2.1 in [17].

4.3 Transitions between standard polynomials and polynomial sumterms

We now turn to the relationship between standard polynomials and polynomial sumterms. Upon evaluation of the numerals that serve as its coefficients, a polynomial sumterm p with variables in $X_1, \ldots, X_n$ can be understood as a standard polynomial p' in the ring $\mathbb{Q}[X_1, \ldots, X_n]$. Thus, we have the translation:

$$p \mapsto p'.$$

Conversely, a polynomial $\alpha \in \mathbb{Q}[X_1, \ldots, X_n]$ can be written as a polynomial sumterm $\alpha^\star$ by turning all coefficients in $\mathbb{Q}$ into the corresponding numerals. Thus, we have the translation:

$$\alpha \mapsto \alpha^\star.$$

Proposition 8. *Given polynomial sumterms p and q involving the same variables and a ring R, the following equivalence holds:*

$$Enl_{\perp}(R) \models p = q \textit{ if, and only if, } p' = q' \textit{ in } R.$$

Moreover, the following observations can be made, which, however, critically depend on the assumption that all coefficients of a polynomial are non-zero – this is because, working in $\mathbb{Q}$, $0 \cdot x = 0$ is true in $\mathbb{Q}$ but not true in $Enl_{\perp}(\mathbb{Q})$.

Proposition 9. *For all polynomials α and β with non-zero coefficients:*

$$\alpha = \beta \textit{ in } R \textit{ if, and only if, } Enl_{\perp}(R) \models \alpha^\star = \beta^\star.$$

Proposition 10. *For all polynomials α and β with non-zero coefficients:*

$$\alpha = \beta \textit{ in } \mathbb{Q} \textit{ if, and only if, } Enl_{\perp}(\mathbb{Q}) \models \alpha^\star = \beta^\star$$
$$\textit{if, and only if, } E_{\mathsf{wcr},\perp} \vdash \alpha^\star = \beta^\star.$$

Proof. This follows by combining Proposition 9 with Proposition 7.

Properties of polynomial sumterms and standard polynomials correspond as follows:

(i) p is non-zero $\Longleftrightarrow p'$ is non-zero,
(ii) p has degree $n \Longleftrightarrow p'$ has degree n,
(iii) p is irreducible $\Longleftrightarrow p'$ is irreducible,
(iv) p is primitive $\Longleftrightarrow p'$ is primitive,
(v) q is a factor of $p \Longleftrightarrow q'$ is a factor of p',
(vi) any polynomial sumterm p can be written as $\underline{a} \cdot q$ for a non-zero integer a and a primitive polynomial sumterm q.

4.4 Quasi-polynomial sumterms

Consider, for instance, the Σ-terms

$$x \text{ and } x + 0 \cdot y.$$

On evaluating in a commutative ring R, these terms over Σ_r define the same functions, but they do not do so in the enlargement $Enl_{\perp}(R)$ as they take different values upon choosing $x = 0, y = \perp$. Thus, the terms need to be distinguished: since 0 usefully occurs as a coefficient in a polynomial when working with $\perp$. We will work with a second kind of polynomial sumterm in order to make these issues explicit.

Definition 32. *A* quasi-polynomial sumterm p *is either*
(i) a polynomial sumterm, or
(ii) a monomial of the form $0 \cdot r$ *with* r *a pure monomial with all its variables occurring in the first power only, or*
(iii) the sum $q + 0 \cdot r$ *of a polynomial sumterm* q *and a monomial of the kind in (ii) and such that no variables commonly occur both in* q *and in* r*.*

The following proposition provides a rationale for the specific form of quasi-polynomial sumterms as just defined.

Proposition 11. *Given a sumterm* p *which contains at least one variable, a pure monomial* q *can be found, with variables occurring with power* 1*, but not with a higher power, such that* $0 \cdot p = 0 \cdot q$*.*

Proof. Let $x_1, \ldots, x_n$ be the variables that occur in p, where these variables are pairwise different. Then take the pure monomial $q = x_1 \cdot \ldots \cdot x_n$.

Note that we need $0 \cdot p = 0 \cdot q$ and not just $p = q$: for example, consider $0 \cdot x \cdot x = 0 \cdot x$ which is an equation valid in all common meadows but does not imply $x \cdot x = x$.

The sum of two polynomial sumterms need *not* be provably equal by $E_{\mathsf{wcr},\perp}$ to a polynomial sumterm. Indeed, $x + (-x) = 0 \cdot x$ is merely a quasi-polynomial sumterm. However, recalling the definitions of 30 and 31, conversely:

Lemma 4. *Using* $E_{\mathsf{wcr},\perp}$*, a product* $r = p \cdot q$ *of two non-zero polynomial sumterms* p *and* q *is provably equal to a polynomial sumterm.*

Proof. We write $t =_{\mathsf{wcr},\bot} r$ for $E_{\mathsf{wcr},\bot} \vdash t = r$. First, notice that $p \cdot q$ is provably equal to a quasi-polynomial sumterm. For example, consider $p \equiv x+1, q \equiv x-1$, then $r = p \cdot q =_{\mathsf{wcr},\bot} (x^2+0\cdot x)+(-1) =_{\mathsf{wcr},\bot} (x\cdot(x+0))+(-1) =_{\mathsf{wcr},\bot} x^2+(-1)$.

More generally, if a variable x occurs in either p or q then as a function $p \cdot q$ depends on x, from which it follows that in the polynomial α with $\alpha = p' \cdot q'$, x must occur at least once in a monomial of α of which the coefficient is non-zero. This implies that an additional summand $0 \cdot x$ is unnecessary in the quasi-polynomial sumterm $\alpha^\star$, which for that reason is provably equal with $E_{\mathsf{wcr},\bot}$ to a polynomial sumterm.

Proposition 12. *Let p and q be integer polynomial sumterms, both with non-zero degree, with variables among $X_1, \ldots, X_n$ and such that p' as well as q' are primitive polynomials.*

Suppose that in $Enl_\bot(\overline{\mathbb{Q}})$ both p and q have value 0 on the same argument vectors in $Enl_\bot(\overline{\mathbb{Q}})^n$. Then, there are

(i) a positive natural number m,

(ii) integer polynomial sumterms $r_1, \ldots, r_m$ with non-zero degree, such that $r'_1, \ldots, r'_n$ are primitive polynomials, and

(iii) non-zero natural numbers $a_1, \ldots, a_n, b_1, \ldots, b_m$ such that

$$E_{\mathsf{wcr},\bot} \vdash p = r_1^{a_1} \cdot \ldots \cdot r_n^{a_m} \text{ and } E_{\mathsf{wcr},\bot} \vdash q = r_1^{b_1} \cdot \ldots \cdot r_n^{b_m}.$$

Proof. Let p and q be as assumed in the statement of the Proposition. Now p and q evaluate to 0 for the same argument vectors in $Enl_\bot(\overline{\mathbb{Q}})^n$. It follows that p and q must contain precisely the same variables. To see this, assume otherwise that say variable x occurs in p and not in q (the other case will work similarly) and then choose a valuation for the other variables in $\overline{\mathbb{Q}}$ which solves $q = 0$, by additionally having value $\bot$ for x a valuation is obtained where $q = 0$ and $p = \bot$, thereby contradicting the assumptions on p and q. Both p' and q' then have non-zero degree and are non-zero polynomials with, using Proposition 8, the same zeroes in $(\overline{\mathbb{Q}})^n$.

Now Proposition 6 can be applied with $\alpha \equiv p', \beta \equiv q'$ thus finding polynomial sumterms $\gamma_1, \ldots, \gamma_m$, and numbers $a_1, \ldots, a_m, b_1, \ldots, b_m$ such that in $\overline{\mathbb{Q}}$:

$$p' = \alpha = \gamma_1^{a_1} \cdot \ldots \cdot \gamma_m^{a_m} \text{ and } q' = \beta = \gamma_1^{b_1} \cdot \ldots \cdot \gamma_m^{b_m}.$$

Now choose: $r_1 \equiv \gamma_1^\star, \ldots, r_m \equiv \gamma_m^\star$. It follows that

$$Enl_\bot(\overline{\mathbb{Q}}) \models p = r_1^{a_1} \cdot \ldots \cdot r_m^{a_m} \text{ and } Enl_\bot(\overline{\mathbb{Q}}) \models q = r_1^{b_1} \cdot \ldots \cdot r_m^{b_m}.$$

Moreover, with Proposition 9, we know that $r_1^{a_1} \cdot \ldots \cdot r_m^{a_m}$ is provably equal to a polynomial sumterm, say P (by $E_{\mathsf{wcr},\bot}$) and that $r_1^{b_1} \cdot \ldots \cdot r_m^{b_m}$ is provably equal to a polynomial sumterm, say $\mathbb{Q}$. So we find $Enl_\bot(\overline{\mathbb{Q}}) \models p = P$ and $Enl_\bot(\overline{\mathbb{Q}}) \models q = Q$, and in consequence $Enl_\bot(\mathbb{Q}) \models p = P$ and $Enl_\bot(\mathbb{Q}) \models q = Q$.

Lastly, using Proposition 7, $E_{\mathsf{wcr},\bot} \vdash p = P$ and $E_{\mathsf{wcr},\bot} \vdash q = Q$ from which one finds that $E_{\mathsf{wcr},\bot} \vdash p = r_1^{a_1} \cdot \ldots \cdot r_m^{a_m}$ and $E_{\mathsf{wcr},\bot} \vdash q = r_1^{b_1} \cdot \ldots \cdot r_m^{b_m}$ thereby completing the proof.

The quasi-polynomial sumterm introduces extra variables via a linear monomial. In [17] extra variables are introduced using a linear sum $0 \cdot (x_1 + \ldots + x_1)$, which takes the same values. From [17] we take the following information concerning sumterms:

Proposition 13. *Let t be a $\Sigma_{r,\perp}$-term, then either*

(i) $E_{\mathsf{wcr},\perp} \vdash t = \perp$; or

(ii) there is a quasi-polynomial sumterm p such that $E_{\mathsf{wcr},\perp} \vdash t = p$.

In each case the reduction is computable.

5 Equational axioms for common meadows

We now add to the equational axioms $E_{\mathsf{wcr},\perp}$ in Table 2 to make a set of equational axioms for common meadows: $E_{\mathsf{ftc-cm}}$ in Table 3. These equations have been presented in a different but equivalent form in [17]. By inspection, one can validate soundness:

Proposition 14. *(Soundness of $E_{\mathsf{ftc-cm}}$.)* $\mathsf{CM} \models E_{\mathsf{ftc-cm}}$.

$$\texttt{import}\ \ E_{\mathsf{wcr},\perp}$$

$$x = \frac{x}{1} \tag{20}$$

$$-\frac{x}{y} = \frac{-x}{y} \tag{21}$$

$$\frac{x}{y} \cdot \frac{u}{v} = \frac{x \cdot u}{y \cdot v} \tag{22}$$

$$\frac{x}{y} + \frac{u}{v} = \frac{(x \cdot v) + (y \cdot u)}{y \cdot v} \tag{23}$$

$$\frac{x}{(\frac{u}{v})} = x \cdot \frac{v \cdot v}{u \cdot v} \tag{24}$$

$$\frac{x}{y + 0 \cdot z} = \frac{x + 0 \cdot z}{y} \tag{25}$$

$$\perp = \frac{1}{0} \tag{26}$$

Table 3. $E_{\mathsf{ftc-cm}}$: Equational axioms for fracterm calculus for common meadows

5.1 On fracterms and flattening

The introduction of division, or a unary inverse, introduces fractional expressions. The theory of fractions is by no means clear-cut if the lack of consensus on their nature is anything to go by [5]. However, in abstract data type theory, fractions can be given a clear formalisation as a syntactic object – as a term over a signature containing $\_/\_$ or $-^{-1}$ with a certain form. Rather than fraction we will speak of a *fracterm*, following the terminology of [5] (item 25 of 4.2).

Definition 33. *A* fracterm *is a term over Σ_{cm} whose leading function symbol is division $\_/\_$. A* flat fracterm *is a fracterm with only one division operator.*

Thus, fracterms have form $\frac{p}{q}$, and flat fracterms have the form $\frac{p}{q}$ in which p and q do not involve any occurrence of division. Note that fracterms are generally defined as terms of the signature Σ_m of meadows, but we will use them only over the Σ_{cm} of common meadows (and its subsignatures). The following simplification process is a fundamental property of working with fracterms.

Theorem 3. *(Fracterm flattening [11].) For each term t over Σ_{cm} there exist p and q terms over Σ_r, i.e., both not involving $\perp$ or division, such that*

$$E_{\mathsf{ftc-cm}} \vdash t = \frac{p}{q},$$

i.e., t is provably equal to a flat fracterm. Furthermore, the transformation is computable.

Proof. Immediate by structural induction on the structure of t, noting that any occurrence of $\perp$ can be replaced by $1/0$.

The set $E_{\mathsf{ftc-cm}}$ of equational axioms for common meadows has been designed so that the proof of fracterm flattening is straightforward; it also allows other results of use for this paper to be obtained easily. More compact but logically equivalent axiomatisations can be found. In [11], using inverse rather than division, a set of logically independent axioms for common meadows is given, from which fracterm flattening is shown, the proof of which then is correspondingly harder.

From now on we will omit brackets thanks to associativity commutativity of addition and multiplication.

5.2 Completeness

We prove that the equations $E_{\mathsf{ftc-cm}}$ are complete for the equational theory CM of common meadows, i.e., for the equational theory of the class of common meadows:

Theorem 4. *For any equation $t = r$ over Σ_{cm} the following holds:*

$E_{\mathsf{ftc-cm}} \vdash t = r$ *if, and only if,* $t = r$ *is valid in all common meadows.*

Proof. The soundness of $E_{\mathsf{ftc-cm}}$ was noted in Proposition 14.

For completeness, suppose that $t = r$ is valid in all common meadows, i.e., $\mathsf{CM} \models t = r$. In what follows, for brevity, we will write $\vdash e$ for $E_{\mathsf{ftc-cm}} \vdash e$.

By the Fracterm Flattening Theorem 3, we can find Σ_r terms p, q, u, v such that

$$\vdash t = \frac{p}{q} \text{ and } \vdash r = \frac{u}{v}.$$

By Proposition 13, each of these four terms can be written in the form of a quasi-polynomial sumterm:

$$\vdash p = s_p + 0 \cdot h_p, \ \vdash q = s_q + 0 \cdot h_q, \ \vdash u = s_u + 0 \cdot h_u, \ \vdash p = s_v + 0 \cdot h_v$$

with s_p, s_q, s_u, s_v polynomial sumterms and h_p, h_q, h_u and h_v linear monomials. Note we don't consider case (i) of 13 because $\perp$ is not a Σ_r term.

Substituting these quasi-polynomial sumterms for p, q, u, v and applying axiom 17 of $E_{\mathsf{ftc-cm}}$, we get

$$\vdash \frac{p}{q} = \frac{(s_p + 0 \cdot h_p) + 0 \cdot h_q}{s_q} \quad \text{and} \quad \vdash \frac{u}{v} = \frac{(s_u + 0 \cdot h_u) + 0 \cdot h_v}{s_v}.$$

So, to prove $\vdash t = r$ we need to prove

$$\vdash \frac{(s_p + 0 \cdot h_p) + 0 \cdot h_q}{s_q} = \frac{(s_u + 0 \cdot h_u) + 0 \cdot h_v}{s_v}$$

assuming its validity in all common meadows.

Now, notice that in all common meadows s_q and s_v must produce 0 on precisely the same non-$\perp$ valuations of the variables occurring in either of both expressions (because otherwise one of the terms would yield $\perp$ while the other does not, making the equality invalid). Six cases will be distinguished, of which the first five are straightforward to deal with:

(i) $s_q \equiv 0$ and $s_v \equiv 0$. Here, trivially

$$\vdash \frac{(s_p + 0 \cdot h_p) + 0 \cdot h_q}{s_q} = \perp = \frac{(s_u + 0 \cdot h_u) + 0 \cdot h_v}{s_v}.$$

(ii) $s_q \equiv 0$ and $s_v \not\equiv 0$. This is not possible because s_q and s_v must produce 0 on the same valuations of variables and if, for a polynomial sumterm h, $h \not\equiv 0$ then it must be that for some common meadow $Enl_\perp(G(\_/\_))$ and valuation σ, we have $Enl_\perp(G(\_/\_)), \sigma \not\models h = 0$.

(iii) The symmetric case $s_q \not\equiv 0$ and $s_v \equiv 0$ is not possible for reasons corresponding with (ii).

(iv) s_q and s_v are both non-zero numerals, say $s_q = \underline{a}$ and $s_v = \underline{b}$. Now, factorisations of a and b both contain the same prime numbers. To see this, otherwise assume that say prime c is a divisor of a while c is not a divisor of b. Then, working in the prime field F_c of characteristic c, s_q takes value 0 while s_v does not, which is an outcome similar to case (ii). The symmetric case that b has a prime factor c which is not a divisor of a works in the same way.

(v) One of s_q and s_v is a non-zero numeral, while the other one contains one or more variables, i.e., has degree 1 or higher. This situation is impossible because in that case the polynomial sumterm of nonzero degree takes both value zero and nonzero (on appropriate arguments) in $\overline{\mathbb{Q}}$ and for that reason also on appropriate non-$\perp$ valuations for $(\overline{\mathbb{Q}})_\perp$.

(vi) Lastly, we are left with the main case that both s_q and s_v are polynomials with non-zero degree. It suffices to prove this equation Θ

$$\frac{s_p + 0 \cdot (h_p + h_q)}{s_q} = \frac{s_u + 0 \cdot (h_u + h_v)}{s_v}$$

from the validity of Θ in all common meadows.

Now, as a first step, chose non-zero integers a and b as follows: a is the gcd of the coefficients of s_q and b is the gcd of the coefficients of s_v. Further, choose polynomial sumterms $\hat{s}_q$ and $\hat{s}_v$ such that

$$\vdash s_q = \underline{a} \cdot \hat{s}_q \text{ and } \vdash s_v = \underline{b} \cdot \hat{s}_v.$$

Next, we show that a and b must have the same prime factors. If not, say c is a prime factor of a but not of b. In the algebraic closure $\overline{F_c}$ of the prime field F_c of characteristic c a solution (i.e., a valuation σ) exists for the equation $s_v - 1 = 0$; this equation must be of non-zero degree as s_v is of non-zero degree.

We find that $\overline{F_c}, \sigma \models \underline{c} = 0$ so that $\overline{F_c}, \sigma \models \underline{a} = 0$, which implies $\overline{F_c}, \sigma \models s_q = 0$. Furthermore, since c is a prime number and c is not a factor of b, then $b \neq 0$ in F_c and $\overline{F_c}, \sigma \models \underline{b} \neq 0$ and $\overline{F_c}, \sigma \models \hat{s}_v = 1$ so that $\overline{F_c}, \sigma \models s_v = \underline{b} \cdot \hat{s}_v \neq 0$, which contradicts the assumptions made above.

Without loss of generality, we may assume that a and b are both positive, and we take an increasing sequence of prime factors $c_1, \ldots, c_k$ with respective positive powers $e_1, \ldots, e_k$ and $f_1, \ldots, f_k$ such that

$$a = c_1^{e_1} \cdot \ldots \cdot c_k^{e_k} \text{ and } b = c_1^{f_1} \cdot \ldots \cdot c_k^{f_k}.$$

The next step is to notice that $\hat{s}_q$ and $\hat{s}_v$ must have the same zero's in $\overline{\mathbb{Q}}$ and to apply Proposition 12 on the polynomial sumterms $\hat{s}_q$ and $\hat{s}_v$, thereby obtaining a sequence of irreducible and primitive polynomials $r_1, \ldots, r_m$ with positive powers $a_1, \ldots, a_m$ and $b_1, \ldots, b_m$ such that

$$\vdash \hat{s}_q = r_1^{a_1} \cdot \ldots \cdot r_m^{a_m} \text{ and } \vdash \hat{s}_v = r_1^{b_1} \cdot \ldots \cdot r_m^{b_m}.$$

By substitutions into equation Θ above, now we know that

$$\vdash \frac{s_p + 0 \cdot (h_p + h_q)}{\underline{a} * \hat{s}_q} = \frac{s_u + 0 \cdot (h_u + h_v)}{\underline{b} * \hat{s}_v}$$

because

$$\vdash s_q = \underline{a} \cdot \hat{s}_q \text{ and } \vdash s_v = \underline{b} \cdot \hat{s}_v.$$

And, proceeding with substitutions, we get:

$$\overline{\mathbb{Q}} \models \frac{s_q + 0 \cdot (h_p + h_q)}{c_1^{e_1} \cdot \ldots \cdot c_k^{e_k} \cdot r_1^{a_1} \cdot \ldots \cdot r_m^{a_m}} = \frac{s_u + 0 \cdot (h_u + h_v)}{c_1^{f_1} \cdot \ldots \cdot c_k^{f_k} \cdot r_1^{b_1} \cdot \ldots \cdot r_m^{b_m}}.$$

It suffices to prove the same equation from $E_{\mathsf{ftc-cm}}$ and to that end we proceed in the following manner.

First, notice by the usual rules of calculation, available from $E_{\mathsf{ftc-cm}}$,

$$\frac{1}{x} = \frac{1+0}{x} = \frac{1}{x} + \frac{0}{x} = \frac{x + 0 \cdot x}{x \cdot x} = \frac{(1+0) \cdot x}{x \cdot x} = \frac{x}{x \cdot x}.$$

Then, let K_{max} be the maximum of $e_1, \ldots, e_k, f_1, \ldots, f_k, a_1, \ldots, a_m, b_1, \ldots, b_m$, and let $K = K_{max} + 1$.

Now, we make repeated use the validity of

$$\frac{x + 0 \cdot w}{y \cdot z^g} = \frac{(x \cdot z^h) + 0 \cdot w}{y \cdot z^{g+h}} \quad (\star)$$

for positive integers g and h (in this case for $g + h = K$) in order to transform the above equation into another, but equivalent, equation between flat fracterms with the same denominator. The identity $(\star)$ is a consequence of the validity of the equations $\frac{1}{x} = \frac{x}{x \cdot x}$ and $(x + (0 \cdot y)) \cdot z = (x \cdot z) + (0 \cdot y))$.

Let

$$\hat{t} \equiv \frac{s_q + 0 \cdot (h_p + h_q)}{c_1^{e_1} \cdot \ldots \cdot c_k^{e_k} \cdot r_1^{a_1} \cdot \ldots \cdot r_m^{a_m}} \text{ and } \hat{r} \equiv \frac{s_u + 0 \cdot (h_u + h_v)}{c_1^{f_1} \cdot \ldots \cdot c_k^{f_k} \cdot r_1^{b_1} \cdot \ldots \cdot r_m^{b_m}}.$$

Moreover, let

$$\hat{\hat{t}} \equiv \frac{(s_q \cdot c_1^{K-e_1} \cdot \ldots \cdot c_k^{K-e_k} \cdot r_1^{K-a_1} \cdot \ldots \cdot r_m^{K-a_m}) + 0 \cdot (h_p + h_q)}{c_1^K \cdot \ldots \cdot c_k^K \cdot r_1^K \cdot \ldots \cdot r_m^K}$$

and

$$\hat{\hat{r}} \equiv \frac{(s_u \cdot c_1^{K-f_1} \cdot \ldots \cdot c_k^{K-f_k} \cdot r_1^{K-b_1} \cdot \ldots \cdot r_m^{K-b_m}) + 0 \cdot (h_u + h_v)}{c_1^K \cdot \ldots \cdot c_k^K \cdot r_1^K \cdot \ldots \cdot r_m^K}.$$

Here it is assumed that the variables in h_q do not occur elsewhere in $\hat{\hat{t}}$ and that the variables of h_u do not occur elsewhere in $\hat{\hat{r}}$; this can be achieved modulo provable equality by means of the equations in $E_{\mathsf{ftc-cm}}$.

With repeated use of the identity $(\star)$ we find that $\vdash \hat{t} = \hat{\hat{t}}$ and $\vdash \hat{r} = \hat{\hat{r}}$.

Summarizing the above, we have established that

$$\vdash t = \hat{t} = \hat{\hat{t}}, \ \ \vdash r = \hat{r} = \hat{\hat{r}} \text{ and } \ Enl_{\perp}(\overline{\mathbb{Q}}) \models \hat{\hat{t}} = \hat{\hat{r}}.$$

Consider the numerators and let

$$H_t = s_q \cdot c_1^{K-e_1} \cdot \ldots \cdot c_k^{K-e_k} \cdot r_1^{K-a_1} \cdot \ldots \cdot r_m^{K-a_m}$$

and
$$H_r = s_u \cdot c_1^{K-f_1} \cdot \ldots \cdot c_k^{K-f_k} \cdot r_1^{K-b_1} \cdot \ldots \cdot r_m^{K-b_m}.$$

Then, from $Enl_{\perp}(\overline{\mathbb{Q}}) \models \hat{\hat{t}} = \hat{\hat{r}}$, it follows that working in $Enl_{\perp}(\mathbb{Q})$ for all non-$\perp$ rational substitutions σ, if $Enl_{\perp}(\overline{\mathbb{Q}}), \sigma \models c_1^K \cdot \ldots \cdot c_k^K \cdot r_1^K \cdot \ldots \cdot r_m^{K-b_m} \neq 0$ it must be the case that $Enl_{\perp}(\overline{\mathbb{Q}}), \sigma \models H_t = H_r$.

So, for all non-$\perp$ valuations σ,
$$Enl_{\perp}(\overline{\mathbb{Q}}), \sigma \models (c_1^K \cdot \ldots \cdot c_k^K \cdot r_1^K \cdot \ldots \cdot r_m^K) \cdot (H_t - H_r) = 0.$$

Rings of polynomials over $\mathbb{Q}$ have no zero divisors and the polynomial sumterm $c_1^K \cdot \ldots \cdot c_k^K \cdot r_1^K \cdot \ldots \cdot r_m^K$ is non-zero. Thus, it follows that, $H_t - H_r = 0$ as polynomials so that $\vdash H_t = H_r$.

Finally, we complete the proof by noticing that
$$\vdash H_t + 0 \cdot (h_p + h_q) = H_r + 0 \cdot (h_u + h_v)$$
because otherwise both terms contain different variables which cannot be the case.

To see this latter point, notice that if, say x occurs in $H_t + 0 \cdot (h_p + h_q)$ and not in $H_r + 0 \cdot (h_u + h_v)$, then, because $H_t = H_r$, a contradiction with $Enl_{\perp}(\overline{\mathbb{Q}}) \models \hat{\hat{t}} = \hat{\hat{r}}$ is arises: contemplate any valuation σ that satisfies $Enl_{\perp}(\overline{\mathbb{Q}}) \models c_1^K \cdot \ldots \cdot c_k^K \cdot r_1^K \cdot \ldots \cdot r_m^{K-b_m} - 1 = 0$, a requirement which is independent of x. Indeed, now the RHS depends on x while the LHS does not, which is a contradiction, thereby completing the proof.

Theorem 5. *The equational theory of common meadows is decidable.*

Proof. Given an equation e, if it is true in all common meadows then it is provable from $E_{\mathsf{ftc-cm}}$. The equations provable from this finite set $E_{\mathsf{ftc-cm}}$ are computably enumerable (Lemma 1). Thus, the true equations of the equational theory of common meadows are computably enumerable. If e is not true in all common meadows then e fails in an algebraic closure of some prime field $\overline{\mathbb{Q}}$ or $\overline{F_p}$ for some prime p. These fields are computable and can be computably enumerated uniformly [51,49], and a computable search for a counterexample to e attempted. Thus, the false equations of the equational theory of common meadows are computably enumerable. In consequence, the equational theory of common meadows is decidable.

Of course, this enumeration argument for decidability is crude. However, we note that the completeness proof for Theorem 4 is effective because the transformations which are used are all computable – including the earlier necessary lemmas such as flattening (Theorem 3) and reductions to quasi-polynomials (Proposition 13). From these transformations, which map the provability of equations to the identity of terms, an alternate proof of decidability can be constructed that offers an algorithm for the provability and validity of equations and invites a further independent analysis.

6 Reflections and prospects

To better appreciate the results of this paper, it maybe helpful to discuss a number of topics in detail: the nature of soundness and completeness theorems (6.1); the origins and development of our research programme (6.2), and its aims and motivation (6.3); and some further technical matters to do with the results (6.4).

6.1 Axioms, calculi and their soundness and completeness

One cannot reason without starting assumptions, and what can be deduced from them by 'sound' logical reasoning will be statements that are true of all contexts where those assumptions apply. In a logical calculus L, mathematical or computational, this means that the theorems that are formally expressed in the language of L, and deduced using its rules, hold true of *all* models of the axioms. This property is called *soundness* and must be proved for each logical calculus L and its chosen semantics S. Conversely, there is the property of *completeness* when any formal statement in the languageof L that is true of the semantics S can be proved using the rules of L. The soundness and completeness of first order logic is the classical example: for any first order theory A and first order logic L, (i) a first order formula ϕ derived from A by L is true of *all* models of A; and (ii) if ϕ is true of *all* models A then is provable in L.

However, it is commonly the case that a set A of axioms has been designed to capture the essential properties of a particular class of models M. This is the case when using axioms to understand number systems, whether with philosophical, mathematical or computational motivations. Any sound calculus that reasons with the axioms in A is *not* talking about the desired target class of models M only, but actually *all* possible models of A. If just one model of A falsifies a statement ϕ then that statement cannot be proved from A. Since the Löwenheim-Skolem Theorem (c.1920), it has been known that first order axiomatisations cannot determine the cardinality of their models. Gödel's Incompleteness Theorem on first order reasoning about natural number arithmetics with the Peano axioms is the classic example of this situation.

Given a set A of axioms, a formal language and proof rules for a logic L, the scope and limits of reasoning are *defined exactly* by soundness and completeness: soundness being necessary to be worth getting started, and completeness being a difficult technical problem if one is interested in a particular subclass of models of the axioms.

In working with data types, this state of affairs is almost standard: it is a common task to seek axioms to analyse the essential properties of the operators of an interface to a *particular* class of semantic models; indeed, the class is often narrowly focussed, being all isomorphic copies of a particular data type that is computable, as is the case with number systems.

Equations are used for axioms as a means of specification, reasoning and computation because they are well understood theoretically and practically. They are (i) familiar and user friendly logical formulae; and have (ii) many general mathematical results that are applicable to computing problems; (iii) practical heuristics and working software tools, based on term rewriting [3,54]. Notable in the case of (ii), is the fact that equations have initial algebra semantics that allow the specification of data types uniquely up to isomorphism. It is 50 years since these equational methods were first applied and developed in computer science [31,41,32]. Equational reasoning about equational specifications is possible in most theorem provers based on formal logics and type theories, certainly those containing first order logic. In addition, there is a wealth of specification and reasoning tools optimised for equations, such as the mature and widely admired Maude [25,26,39], and its predecessors and successors.

The class of commutative rings with 1 are defined by finitely many equations (Table 1). However, the class of all fields is first order requiring the addition of a Π_1^0 or $\forall$-formula. Furthermore, as we noted earlier, in the origins of the algebraic methods [21], there is:

Birkhoff's Theorem. *Let* K *be a class of* Σ*-algebras. Then* K *has an equational axiomatisation* E *if, and only if, the class* K *is closed under subalgebras, homomorphic images and products.*

From this it follows that the class of all fields, and the class of all common meadows, cannot be defined by equations, as these classes are not closed under products.

There are different semantic models of the data type/meadow of rational numbers $\mathbb{Q}$ that we will note in the next section, all of which have been given equational specifications under initial algebra semantics. In each case, the equational calculi that are used to reason with the equational specifications are talking about far more than the rationals. In particular, in the case of common meadows, our completeness theorem here answers the question as to what the axioms of $E_{\mathsf{ftc-cm}}$ can actually talk about using the language of equations only, namely: *all fields with the error value* $\perp$. This confirms the significance of $E_{\mathsf{ftc-cm}}$ and its equivalents.

So, in proving mathematical properties – say, of logics to prove that programs meet specifications – axioms and calculi are necessary. Issues of partiality versus totality for operations arise and must be addressed for both computational and logical reasons. Partiality can play a valuable role in specifications. There are different interpretations of partiality in addition to the orthodox 'no element' in a specification: partiality can simply stand for some element yet to be defined, possibly one that is quite arbitrary. For example, partiality is an important semantic feature of the algebraic specification language CASL [23].

In any practical computing system, invoking an operator *must* return some value or message: returning no value leads to an unacceptable indefinite wait for the user. This means that data types cannot have partial operations when they are to be implemented. In formal reasoning, to make logical methods that are tractable and robust for implementing usable software tools, eliminating terms that have no meaning is necessary.

Thus, where data types involve numbers, division must be totalised by some semantic choice for $\frac{x}{0}$. This has long been done in calculators, languages and theorem proving (e.g., in theorem provers, such as Coq and Lean).

Our many results on this question point to $\perp$ and common meadows as the best choice.

6.2 Semantical options for the problem of division by zero

Completely central to quantification and computation are the rational numbers $\mathbb{Q}$. When we measure the world using a system of units and subunits then we use the rational numbers. Today's computers calculate only within subsets of the rational numbers. An early motivation for our theory is to design and analyse abstract data types of rational numbers. Designing a data type for rationals requires algebraic minimality, which can be obtained by introducing either division or inverse as an operation. Thus, division is essential for data type of rational numbers and must be total, which requires choosing a value for $1/0$.

Using various semantical values to be found in practical computations to totalise division – such as error, ∞, NaN, the last standing for 'not a number' – we have constructed equational specifications (under initial algebra semantics) for the following data types of rational numbers:

Involutive meadows, where an element of the meadow's domain is used for totalisation, in particular $1/0 = 0$, [14];

Common meadows, the subject of this paper, where a new external element $\perp$ that is 'absorbtive' is used for totalisation $1/0 = \perp$, [11];

Wheels, where one external ∞ is used for totalisation $1/0 = \infty = -1/0$, together with an additional external error element $\perp$ to help control the side effects of infinity, [48,24,16];

Transrationals, where besides the error element $\perp$ two external signed infinities are added, one positive and one negative, so that division is totalised by setting $1/0 = \infty$ and $-1/0 = -\infty$, [1,28,15];

In practice, the first four of these models are based on data type conventions to be found in theorem provers, common calculators, exact numerical computation and, of course, floating point computation, respectively. A fifth, the symmetric transrationals, that we developed is discussed in the next section.

For some historical remarks on division by zero, we mention [2], and for a survey we mention [4].

Of these semantical options it may be helpful to compare the common meadows with at least one of the above. The simplest and most common choice *appears* to be the involutive meadows, which have been deployed in logical arguments and has its advocates [43,44]. In our [14], to create an equational specification for the rational numbers, we introduced totality by setting $0^{-1} = 0$. This led us to the study of involutive meadows [14,8,9], and subsequently to the broad programme of work mentioned earlier.

An explicit logical discussion of the proposal to adopt $0^{-1} = 0$ dates back at least to Suppes [53], and to theoretical work of Ono [45]. A completeness result was shown by Ono [45]. In [8], the equational theory of involutive meadows was introduced. Completeness for the Suppes-Ono equational theory is shown with a different proof in [6]. An advantage of the latter approach to completeness is that it generalises to the case of ordered meadows, see also [7].

Although the flattening property is quite familiar from the school algebra of rational numbers, it validity for common meadows (Theorem 3) stands in marked contrast with the abstract situation for involutive meadows. In [6] it is shown that, with the axioms for involutive meadows, terms are provably equal to only *finite sums* of flat fracterms; and in [10], it is shown that *arbitrarily large numbers of summands of flat fracterms* are needed for that purpose. Thus, the involutive meadows run into fundamental algebraic difficulties that the common meadows do not.

Our results here and elsewhere point to the fact that arithmetical abstract data types with error values are theoretically superior among the many practical conventions we have studied. This design decision is attractive semantically since $\perp$ as an error value can have different interpretations in computations. Futhermore, much of the algebra we have encountered for common meadows is intimately and agreeably connected with the theories of rings and fields; and it serves rather well the theory of data types of rational numbers, which must be a starting point for theorising.

6.3 Aims of the programme

This division by zero problem proved to be the beginning of a 15 year programme of algebraic research into computer arithmetics. Clearly, as discussed in the previous section, one of the aims of our research programme is to discover the mathematical implications of some of the different options of totalising division in number systems, and to establish an understanding of what might be best practice. These results on specifications have a role in designing, new more mathematically and logically useful specifications of computer arithmetics.

Floating point systems exhibit a number of pathologies [34] and are very complicated to analyse and reason about [46,47]. Thus, over decades there has been renewed interest in creating computer arithmetics distinct from floating point. Alternate models have been designed for ‘exact computations’ with real numbers, firmly focused on working implementations rather than on their algebra, specification and reasoning. An important early example is Interval Analysis that works with intervals rather than points to accommodate errors in measurements or due to rounding [42]; for an introduction to Interval Analysis, see [55]. In the 1990s, there was a resurgence of interest in computable analysis and topology, which also led to quite distinct semantical models and implementations, having distinct goals and aspirations. For the purpose of computability theory, the different models were equivalent – see the partial survey [52].

The common meadow has led us to re-examine, from our algebraic point of view, further general properties that are seen in computer arithmetics, old and new. Building on the common meadow, we have designed and specified a new data type of rational numbers in [19], motivated by some common floating point conventions. Called the *symmetric transrationals*, the data type employs the error element $\perp$, two external signed infinities $+\infty, -\infty$, and two infinitesimals $+\iota, -\iota$ so that division is totalised by setting $1/0 = \perp$, as with common meadows; and the other elements are used to manage overflows and underflows. The symmetric transrationals implement these three features:

(i) having total operations only;

(ii) accommodating overflows and underflows with respect to upper and lower numerical bounds; and

(iii) computations are sensitive when values come close to 0.

In particular, it separates totality from over- and underflows.

Our programme continues with other topics are being addressed, including: (i) partiality in abstract data types and term rewriting; (ii) the effect of the different semantics on the computational power of imperative programs on arithmetic structures; (iii) the interpretation of fractions and their pragmatics in teaching; and (iii) the connection between various meadows and advanced ring theory (such as von Neumann rings). In fact, our main theorem here has been applied in the proof of a theorem on term rewriting in [20].

6.4 Matters arising

Being close to the axioms for commutative rings, the axioms $E_{\mathsf{ftc-cm}}$ are not unfamiliar and hopefully memorable. The equational axiomatisation $E_{\mathsf{ftc-cm}}$ has been optimised for ease of use in the paper (e.g., especially flattening), and we have not paid attention to the logical independence of the various axioms. Some of the axioms of $E_{\mathsf{ftc-cm}}$ are redundant, given the other ones. Given their arithmetic purpose, the relationships between axiomatisations of common meadows and axiomatisations of rings and fields are of mathematical interest and practical value. Finding attractive sets of axioms which are also minimal is a topic worthy of investigation in its own right. In the revision of [11] the same equational theory, though equipped with inverse rather than with division, is given an axiomatisation with logically independent axioms.

Three open questions stand out from the results in this paper:

(i) Is the equational theory of the common meadow $Enl_\perp(\mathbb{Q}(\_/\_))$ of rationals decidable?

(ii) Can a finite basis for the equational theory of common meadows with orderings be found?

(iii) Can the equational theory of common meadows be axiomatised by means of a specification which constitutes a complete term rewriting system?

In the matter of (i), we know two things. First, that our $E_{\mathsf{ftc-cm}}$ is *not* complete for the equational theory of $Enl_\perp(\mathbb{Q}(\_/\_))$. For instance, the equation $(X^2+1)/(X^2+1) = 1$ is true in $Enl_\perp(\mathbb{Q}(\_/\_))$ but it is wrong in a meadow of complex numbers $Enl_\perp(\mathbb{C}(\_/\_))$, where $x = i$ is possible; thus, it cannot be derivable from $E_{\mathsf{ftc-cm}}$ which is sound for all common meadows, including the common meadow obtained from complex numbers. Second, we have shown that the equational theory of the common meadow of rational numbers has the 1-1 degree of the Diophantine Problem for rational numbers, which is an important long-standing open problem, see [18].

In the matter of (ii), this was done in the setting of $1/0 = 0$ using a sign function in [6].

In the matter of (iii), a negative result in a simplified case was obtained in [12].

Notwithstanding these open questions, we consider common meadows to provide an attractive starting point for the algebraic and logical study of a practical semantics for reasoning about arithmetical data types.

References

1. James A. Anderson, Norbert Völker, and Andrew A. Adams. 2007. Perspecx Machine VIII, axioms of transreal arithmetic. In J. Latecki, D. M. Mount and A. Y. Wu (eds), *Proc. SPIE 6499. Vision Geometry XV*, 649902, 2007.
2. James A. Anderson and Jan A. Bergstra. 2021. Review of Suppes 1957 proposals for division by zero. *Transmathematica*, (2021). `https://doi.org/10.36285/tm.53`.
3. Franz Baader and Tobias Nipkow. 1998. *Term Rewriting and All That.* Cambridge University Press, 1998. `https://doi:10.1017/CBO9781139172752`
4. Jan A. Bergstra. 2019. Division by zero, a survey of options. *Transmathematica*, (2019). `https://doi.org/10.36285/tm.v0i0.17`.
5. Jan A. Bergstra. 2020. Arithmetical data types, fracterms, and the fraction definition problem. *Transmathematica*, (2020). `https://doi.org/10.36285/tm.33`.
6. Jan A. Bergstra, Inge Bethke and Alban Ponse. 2013. Cancellation meadows: a generic basis theorem and some applications. *The Computer Journal*, 56 (1) (2013), 3–14. Also `arxiv.org/abs/0803.3969`.
7. Jan A. Bergstra, I. Bethke, and A. Ponse. 2015. Equations for formally real meadows. *Journal of Applied Logic*, 13 (2) (2015), 1–23.
8. Jan A. Bergstra, Yoram Hirshfeld, and John V. Tucker. 2009. Meadows and the equational specification of division. *Theoretical Computer Science*, 410 (12) (2009), 1261–1271.
9. Jan A. Bergstra and C.A. Middelburg. 2015. Division by zero in non-involutive meadows. *Journal of Applied Logic*, 13(1): 1–12 (2015). `https://doi.org/10.1016/j.jal.2014.10.001`
10. Jan A. Bergstra and Cornelis A. Middelburg. 2015. Transformation of fractions into simple fractions in divisive meadows. *Journal of Applied Logic*, 16 (2015), 92–110. Also `https://arxiv.org/abs/1510.06233`.
11. Jan A. Bergstra and Alban Ponse. 2015. Division by zero in common meadows. In R. de Nicola and R. Hennicker (eds), *Software, Services, and Systems: Wirsing Festschrift*, Lecture Notes in Computer Science 8950, Springer, 2015, 46–61. For an improved version (2021), see: `arXiv:1406.6878v4`.
12. Jan A. Bergstra and Alban Ponse. 2016. Fracpairs and fractions over a reduced commutative ring. *Indigationes Mathematicae*, 27, (2016), 727–748. Also `https://arxiv.org/abs/1411.4410`.
13. Jan A. Bergstra and J.V. Tucker. 1995. Equational specifications, complete term rewriting systems, and computable and semicomputable algebras. *Journal of the ACM*, Vol. 42 (6), 1194-1230 (1995).
14. Jan A. Bergstra and John V. Tucker. 2007. The rational numbers as an abstract data type. *Journal of the ACM*, 54 (2) (2007), Article 7.

15. Jan A. Bergstra and John V. Tucker. 2020. The transrational numbers as an abstract data type. *Transmathematica*, (2020). `https://doi.org/10.36285/tm.47`.
16. Jan A. Bergstra and John V. Tucker. 2021. The wheel of rational numbers as an abstract data type. In Roggenbach M. (editor), *Recent Trends in Algebraic Development Techniques. WADT 2020*. Lecture Notes in Computer Science 12669, Springer, 2021, 13–30.
17. Jan A. Bergstra and John V. Tucker. 2022. On the axioms of common meadows: Fracterm calculus, flattening and incompleteness. *The Computer Journal*. Online first, 8pp. `https://doi.org/10.1093/comjnl/bxac026`
18. Jan A. Bergstra and John V. Tucker. 2022. Totalising partial algebras: Teams and splinters. *Transmathematica*, `https://doi.org/10.36285/tm.57`
19. Jan A. Bergstra and John V. Tucker. 2022. Symmetric transrationals: The data type and the algorithmic degree of its equational theory, in N. Jansen et al. (eds.) *A Journey From Process Algebra via Timed Automata to Model Learning - A Festschrift Dedicated to Frits Vaandrager on the Occasion of His 60th Birthday*, Lecture Notes in Computer Science 13560, 63-80. Springer, 2022.
20. Jan A. Bergstra and John V. Tucker. 2023. Eager term rewriting for the fracterm calculus of common meadows, *The Computer Journal*, 2023. bxad106. `https://doi.org/10.1093/comjnl/bxad106`
21. Garrett Birkhoff. 1935. On the Structure of Abstract Algebras. *Mathematical Proceedings of the Cambridge Philosophical Society*, 31 (4), 433-454.
22. Garrett Birkhoff and Saunders MacLane. 1965. *Survey of Modern Algebra*. Macmillan, 1965.
23. Michel Bidoit and Peter D Mosses. 2004. *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`
24. Jesper Carlström. 2004. Wheels – on division by zero, *Mathematical Structures in Computer Science*, 14 (1), (2004), 143-184.
25. Manuel Clavel, Francisco Durn, Steven Eker, Patrick Lincoln, Narciso Mart-Oliet, Jos Meseguer, J.F. Quesada. 2002. Maude: specification and programming in rewriting logic, *Theoretical Computer Science*, 285 (2), 2002,187-243, `https://doi.org/10.1016/S0304-3975(01)00359-0`.
26. Manuel Clavel, Francisco Durn, Steven Eker, Patrick Lincoln, Narciso Mart-Oliet, Jos Meseguer, and Carolyn L Talcott. 2007. *All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic*, Lecture Notes in Computer Science 4350, Springer, 2007. `https://doi.org/10.1007/978-3-540-71999-1_18`.
27. Claude Chevalley. 1956. *Fundamental Concepts of Algebra*. Academic Press, 1956.
28. Tiago S. dos Reis, Walter Gomide, and James A. Anderson. 2016. Construction of the transreal numbers and algebraic transfields. *IAENG International Journal of Applied Mathematics*, 46 (1) (2016), 11–23. `http://www.iaeng.org/IJAM/issues_v46/issue_1/IJAM_46_1_03.pdf`
29. Hans-Dieter Ehrich, Markus Wolf, and Jacques Loeckx. 1997. *Specification of Abstract Data Types*. Vieweg Teubner, 1997.
30. H. Ehrig and B. Mahr. 1985. *Fundamentals of Algebraic Specification 1: Equations und Initial Semantics*, EATCS Monographs on Theoretical Computer Science, Vol. 6, Springer, 1985.
31. Joseph Goguen, James Thatcher and Eric Wagner. 1976. An Initial Algebra Approach to the Specification, Correctness and Implementation of Abstract Data

Types. Technical Report RC 6487, IBM T. J. Watson Research Center, October, 1976. Reprinted in: Raymond Yeh (editor), *Current Trends in Programming Methodology, IV*. Prentice-Hall, 1978, 80-149.
32. Joseph A. Goguen. 1989. Memories of ADJ. 1989. *Bulletin of the EATCS* no. 36, October 1989. (Available at `https://cseweb.ucsd.edu/~goguen/pps/beatcs-adj.ps`).
33. Wilfrid Hodges. 1993. *Model Theory*. Cambridge University Press, 1993.
34. William Kahan. 2011. *Desperately Needed Remedies for the Undebuggability of Large Floating-Point Computations in Science and Engineering* `https://people.eecs.berkeley.edu/~wkahan/Boulder.pdf`
35. John von Neumann and Hermann Goldstine. 1947. Numerical inverting of matrices of high order. 1947. *Bulletin American Mathematical Society*, 53 (11), 1021-1099.
36. Serge Lang. 1965. *Algebra*. Addison Wesley, 1965.
37. Serge Lang. 2002. *Algebra*. Graduate Texts in Mathematics, Vol. 211, Third revised edition, 2002. Springer.
38. A.I. Mal'tsev. 1973. *Algebraic systems*. Springer-Verlag, 1973.
39. *The Maude System*. `http://maude.cs.illinois.edu/w/index.php/The_Maude_System`
40. K. Meinke and J. V. Tucker. 1992. Universal Algebra. In S Abramsky and D Gabbay and T Maibaum, *Handbook of Logic for Computer Science*, Oxford University Press, 1992, 189–411.
41. Jose Meseguer and Joseph Goguen. 1985. Initiality, induction and computability. In Maurice Nivat and John C. Reynolds (editors), *Algebraic Methods in Semantics* Cambridge University Press, 1985, 459-541.
42. Ramon E Moore. 1966. *Interval Analysis*. Prentice-Hall, 1966.
43. Hiroshi Okumura, Saburou Saitoh and Tsutomu Matsuura. 2017. Relations of zero and ∞. *Journal of Technology and Social Science*, (2017) 1 (1).
44. Hiroshi Okumura. 2018. Is it really impossible to divide by zero? *Biostatistics and Biometrics Open Acc. J.* 7 (1) 555703. DOI: 10.19080/BBOJ.2018.07.555703, (2018)
45. Hiroakira Ono. 1983. Equational theories and universal theories of fields. *Journal of the Mathematical Society of Japan*, 35 (2) (1983), 289-306.
46. M. Overton. 2001. *Numerical Computing with IEEE Floating Point Arithmetic*. SIAM, 2001.
47. S Rump. 2010. Verification methods: Rigorous results using floating-point arithmetic. *Acta Numerica*,19, 287-449. doi:10.1017/S096249291000005X
48. Anton Setzer. 1997. Wheels (Draft), Unpublished. 1997.
49. Albrecht Fröhlich and John C. Shepherdson Effective procedures in field theory, 1956. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences* 248 (1956), 407-432 `http://doi.org/10.1098/rsta.1956.0003`
50. Ian Stewart. 1972. *Galois Theory*. Chapman Hall, 1972.
51. Viggo Stoltenberg-Hansen and John V. Tucker. 1999. Computable rings and fields, in Edward Griffor (ed), *Handbook of Computability Theory*, Elsevier, 1999, 363-447.
52. Viggo Stoltenberg-Hansen and John V. Tucker. Concrete models of computation for topological algebras, *Theoretical Computer Science*, 219 (1999) 347-378 https://doi.org/10.1016/S0304-3975(98)00296-5
53. Patrick Suppes. 1957. *Introduction to Logic*. Van Nostrand Reinhold, 1957.
54. Terese, *Term Rewriting Systems*. Cambridge Tracts in Theoretical Computer Science 55. Cambridge University Press, 2003.
55. Warwick Tucker. 2011. *Validated Numerics: A Short Introduction to Rigorous Computations'*. Princeton University Press, 2011.

56. John V Tucker. 2022. Unfinished Business: Abstract data types and computer arithmetic. *BCS FACS FACTS*, The Newsletter of the Formal Aspects of Computing Science BCS Specialist Group, Issue 2022-1, February 2022, 60-68. `https://www.bcs.org/media/8289/facs-jan22.pdf`
57. B L van der Waerden. *Modern Algebra. Volume 1.* Frederick Ungar Publishing Company, 1970.
58. Wolfgang Wechler. *Universal Algebra for Computer Scientists.* Springer-Verlag, 1992.