Totalising Partial Algebras

Teams and Splinters

Bergstra, J.A.; Tucker, J.V.

DOI
10.36285/tm.57

Publication date
2022

Document Version
Final published version

Published in
Transmathematica

Transmathematica DOI Prefix 10.36285 ISSN 2632-9212

Totalising Partial Algebras: Teams and Splinters

Jan A. Bergstra
j.a.bergstra@uva.nl janaldertb@gmail.com
Informatics Institute, University of Amsterdam,
Science Park 904, 1098 XH, Amsterdam,
The Netherlands

John V. Tucker
j.v.tucker@swansea.ac.uk
Department of Computer Science,
Swansea University, Bay Campus,
Fabian Way, Swansea, United Kingdom, SA1 8EN

Submitted: 17 April 2021
Revised: 27 March 2022

Abstract

We will examine totalising a partial operation in a general algebra by using an absorbtive element $\perp$, such as an error flag. We then focus on the simplest example of a partial operation, namely subtraction on the natural numbers: $n - m$ is undefined whenever $n < m$. We examine the use of $\perp$ in algebraic structures for the natural numbers, especially semigroups and semirings. We axiomatise this totalisation process and introduce the algebraic concept of a *team*, being an additive cancellative semigroup with totalised subtraction. Also, with the natural numbers in mind, we introduce the property of being generated by an iterative function, which we call a *splinter*. We prove a number of theorems about the algebraic specification of datatypes of natural numbers.

1 Introduction

We will examine totalising a partial operation in a general algebra by introducing a special element to flag when the operation is not defined. The process involves adding a new element $\perp$ to the algebra, and re-defining each operation f of the algebra on its arguments x such that

(a) if x does not contain $\perp$ and $f(x)$ is undefined then $f(x) = \perp$; and

(b) if x does contain $\perp$ then $f(x) = \perp$, i.e., $\perp$ is *absorbtive*.

Augmenting a partial structure with an absorbing flag like $\perp$ is familiar in practice. In the arithmetical structures to be found in calculators, the absorbtive element $\perp$ is the error flag, returned as the value for $1/0$, $tan(\pi/2)$, $log(0)$ etc. Modeling the output of a calculator on say $1/0$ constitutes a simplification, however, which may fail to take into account that often the result cannot be used as an input for further calculation, an option which one might expect or even require for $\perp$. Connections between $\perp$ and actual computational phenomena cannot be taken for granted and will require detailed scrutiny depending on the case at hand.

An advantage of totalisation, which primarily motivates our work, is that it simplifies (massively!) logical reasoning.

What effect does introducing this simple absorbtive element into a partial algebra, for the purposes of totalisation, have on the operations of an algebra and their standard properties?

We will consider this question in a general case using the theory of abstract datatypes. First, we describe the totalisation method for datatypes in detail, and establish the effect of applying the totalisation method in the task of equationally specifying a datatype.

Our main interest is in arithmetical structures. Here, we will study simple examples of arithmetical structures built on a set $\mathbb{N}$ of natural numbers. The standard algebraic structures on the natural numbers are total: an additive semigroup; a multiplicative semigroup; and, by gluing these together with distribution laws, semirings; the semigroups and semirings are commutative. However, there are many more structures: we begin with studying the totalisation of what must be the simplest partial operation, namely subtraction on a set $\mathbb{N}$ of natural numbers:

$$n - m = \perp \text{ whenever } n, m \in \mathbb{N} \text{ and } n < m.$$

As we will see, the totalisation method is dependent on what structures are built on the naturals.

Abstractly, subtraction on the natural numbers is viewed through the lens of an additive semigroup structure. We show how an additive semigroup can have a partial subtraction operation that can be made total by enriching the semigroup with $\perp$. The new structure is something between an additive semigroup and an additive group (in which binary subtraction can be defined as a total operation using the group's unary inverse); we call it a *team* – a name pointing in the direction of groups. Notice that a unary additive inverse $-n$ is of no interest on $\mathbb{N}$ since $n + m = 0 \iff n = m = 0$. Specifically, a team is a cancellative

semigroup equipped with a partial subtraction operation made total with $\perp$. We give a first order axiomatisation *Teams* of teams using 9 equations and one implication with a negation as a premiss. We classify the models of the axioms and show:

Theorem 1.1. *Let A be any algebra. Then: $A \models Teams$ if, and only if, A is an enrichment, by subtraction made total by an absorbtive element $\perp$, of an cancellative commutative semigroup.*

The class of all Teams cannot be defined by any set of conditional equations (i.e., the class of teams is not a quasivariety).

For an algebra to qualify as a datatype it must be minimal, i.e., all its elements can be constructed by applying its operations to its constants. The team

$$N_{team} = (\mathbb{N}|0, +, -, \perp)$$

of natural numbers is not minimal. Now, since the work on the natural numbers by Dedekind and Peano it has become conventional to view natural numbers as made from 0 by repeated application of a successor function $n+1$ (although both Dedekind and Peano used 1 as the starting point). Algebras which can be exhausted by repeatedly applying a single unary function are ubiquitous but, unlike semigroups, groups, semirings, rings and fields, have not acquired a dedicated name. We will refer to such structures as *splinters*, using terminology borrowed from the theory of computable functions. In the general algebraic case, we require the iterative generating functions to have explicit definitions in terms of the algebra's operations. Being a splinter is a structural property of algebras that implies minimality.

We apply the general notion of a splinter to our algebras of natural numbers. Of particular interest, is an algebra

$$N_{team,1,\cdot} = (\mathbb{N}|0, 1, \perp, +, -, \cdot)$$

of natural numbers that forms a team under addition, but also has multiplication and a unit. The algebra is an adaptation of a semiring to accommodate subtraction: we refer to it as a *common semiring with subtraction*. The presence of addition and 1 implies the algebra is a splinter. We prove:

Theorem 1.2. *The equational theory of $N_{team,1,\cdot} = (\mathbb{N}|0, 1, \perp, +, -, \cdot)$ does not have a finite basis.*

However, on removing multiplication and leaving 1:

Theorem 1.3. *The equational theory of the team $N_{team,1} = (\mathbb{N}|0, 1, \perp, +, -)$ with 1 is decidable.*

The structure of the paper is this. In Section 2, we explain the method of totalising with $\perp$ for general algebras. In Section 3, we focus on the natural numbers and semigroups and introduce teams. In Section 4, we introduce splinters. In Sections 5 and 6 we examine splinters of natural numbers, $N_{team,1,\cdot}$

and $N_{team,1}$, respectively. In Section 7, we conclude with some remarks about totalisation, subtraction and arithmetical systems.

Preliminaries. We assume the reader is familiar with the basic theory of abstract datatypes and their equational specification: notions of signature, algebra, minimal algebra, terms, equations, conditional equations, morphisms, initial algebras. See, for instance, [16].

A datatype is a minimal algebra and may be total or partial. An abstract datatype is the isomorphism class of a datatype. We will maintain a notational distinction between concrete datatypes and the corresponding abstract datatypes.

We will meet several different datatypes of natural numbers, but we will consistently use the notational conventions:

1. $\mathbb{N}$ for some chosen set of natural numbers;
2. N for some chosen algebraic structure built on the set $\mathbb{N}$ of natural numbers; and
3. $\mathbf{N}$ for the isomorphism class of the algebraic structure N built on the set $\mathbb{N}$ of natural numbers.

If N is a datatype then $\mathbf{N}$ is the abstract datatype to which it belongs.

Let K be a class of Σ-algebras. The *equational theory* $Eqn(K)$ of K is the set of all equations holding true in every algebra belonging to K. A set B of equations is a *basis* for an equational theory E if, and only if, B and E have exactly the same models; E is *finitely based* if E has a finite basis.

Other notions will be introduced *in situ*.

1.1 Impact on computability

Computability of datatypes is not our theme in this paper but some remarks about that matter may be of use. If one starts out with a computable datatype involving total functions only the resulting datatype is computable. For a defintion of computability of datatypes with total functions only we refer to [10]. If one starts out with a datatype A involving partial functions then defining computability of A requires some care. A satisfactory definition of computability for datatypes involving partial functions, is obtained by requiring that A is computable if, and only if, the result of totalizing all operations with a new absorptive element $\perp$ is computable.

If, however, one starts out with a semi-computable (though not computable) datatype, involving partial functions the result of totalising with an absorptive element may (but need not) be a datatype which is not semi-computable. In the setting of elemenatry arithmetic semi-computable datatypes are less common, mainly because semi-computable fields are necessarily also computable. However, even if semi-computablity is lost, the simplification of the logic obtained by totalisation may well be worth the effort.

2 Structures with an absorptive element

We will discuss total and partial structures with and without absorptive elements, focusing on single sorted algebras.

2.1 Single-sorted algebras with absorption

For single sorted signature Σ and Σ-algebra A with carrier set A_s, the definition of an absorptive element is as follows:

Definition 2.1. *A single-sorted Σ-structure A has an* absorptive element $a \in A_s$ *for its sort s if for each operation f of Σ, if one of its arguments equals a then so does its value.*

The following partial result is relevant for arithmetical datatypes, which conventionally are single-sorted and are equipped with a two place addition function.

Proposition 2.1. *If Σ is single sorted and there is at least one function f in Σ with two or more arguments then an absorptive element in a Σ-algebra is unique.*

Proof. Let f have arguments $x_1, x_2, \ldots, x_{n+2}$, $n \geq 0$. Let a and b be absorptive elements of Σ structure A then $a = f(a, b, \ldots, b) = b$. □

Let $\bot$ be a standard notation for an absorbtive element. Thus, normally, our use of $\bot$ in an algebra will imply that it is absorbtive with respect to all the operations of the algebra.

Definition 2.2. *Let A be a single sorted Σ-algebra with sort s and assume $\bot \notin A_s$. Then the* $\bot$-enlargement *of A is an algebra $A_\bot$ with signature $\Sigma_\bot = \Sigma \cup \{\bot : s\}$ and carrier set $(A_\bot)_s = A_s \cup \{\bot\}$, which is an enlargement of A in which $\bot$ is absorptive.*

As a notation for the construction of the $\bot$-enlargement $A_\bot$ from A we will use $\mathsf{Enl}_\bot(A)$. For any class K of Σ-algebras, define

$$\mathsf{Enl}_\bot(K) = \{\mathsf{Enl}_\bot(A) | A \in K\}.$$

Proposition 2.2. *Let Σ be a single-sorted signature which contains at least one function symbol with two or more arguments. For any nonempty class K of Σ-algebras, the class $\mathsf{Enl}_\bot(K)$ cannot be defined by conditional equations, i.e., is not a quasivariety.*

Proof. Let $f \in \Sigma$, and without loss of generality, we assume that f has arity 3. Assume for a contradiction, E is a conditional equational theory over $\Sigma_\bot$ such that $Alg(\Sigma_\bot, E) = \mathsf{Enl}_\bot(K)$.

We notice that $\mathsf{Enl}_\bot(K) \models f(x, y, y) = \bot \to x = \bot \vee y = \bot$, and thus also $E \vdash f(x, y, y) = \bot \to x = \bot \vee y = \bot$. Now if a conditional equational theory proves a disjunction then it must prove either of the two disjuncts.

Suppose $E \vdash f(x,y,y) = \bot \to x = \bot$, the first disjunction; the argument for the other disjunction us similar. Then taking $y = \bot$, and using $E \vdash f(x,\bot,z) = \bot$, which must be the case because it holds in $\mathsf{Enl}_\bot(K)$, we find $E \vdash x = \bot$.

Now consider any algebra A in K, this algebra has a nonempty domain containing at least one value d. We have $d \in (\mathsf{Enl}_\bot(A))_s$ and $d \neq \bot$, by definition of $\mathsf{Enl}_\bot$. So $\mathsf{Enl}_\bot(A) \models x = \bot$ fails and $E \vdash x = \bot$ fails as well, thus arriving at a contradiction that completes the proof. □

In contrast, if only unary functions are present then a counterpart of Proposition 2.2 cannot be shown as axioms $f(x) = \bot \to x = \bot$ do the job.

It follows from Proposition 2.2 that in most cases there is no way to find a complete (conditional) equational specification for $\mathsf{Enl}_\bot(K)$. In particular:

Corollary 2.1. *Given a single-sorted signature Σ which contains at least one function symbol with two or more arguments, and any equationally defined class $K = Alg(\Sigma, E)$, then the class $\mathsf{Enl}_\bot(K)$ is not definable by conditional equations.*

Rather than looking for equational axioms for $\mathsf{Enl}_\bot(K)$, what can be done instead, is to look for a finite basis of the full equational theory of $\mathsf{Enl}_\bot(K)$. At present we do not know under which conditions such a finite basis can be found. More specifically, we have no answer to the following question:

Problem 2.1. *Let E be a finite set of equations over signature Σ. Must it be the case that the equational theory of $\mathsf{Enl}_\bot(Alg(\Sigma, E))$ has a finite basis?*

For a class K of total algebras considering the class $\mathsf{Enl}_\bot(K)$ provides no significant advantages. This fact is illustrated below in Theorem 2.2. The situation changes, however, if K contains partial algebras. We will see later, in subsection 2.3, that the construction $\mathsf{Enl}_\bot(A)$ can easily be enhanced to totalise partial algebras. For a class K involving partial algebras considering the (conditional) equational logic of $\mathsf{Enl}_\bot(K)$ may serve as a substitute for considering the intricate logics of partial functions that apply to the algebras of K.

2.2 Algebraic specifications in the presence of absorption

For a datatype A we write $\mathsf{ADT}(A)$ for the abstract datatype which contains A.

Proposition 2.3. *Let Σ be a single sorted signature including a commutative two place function $+$ with a zero 0 which is also a constant. Assume that (Σ, E) is a finite equational initial algebra specification of $\mathsf{ADT}(A)$. Then $A_\bot(\hat{\rho})$ has a finite equational initial algebra specification as well, with auxiliary function*

$$\hat{\rho} : A_s \cup \{\bot\} \to A_s \cup \{\bot\}$$

such that $\hat{\rho}(x) = 0$ on all $x \in A_s$ and $\hat{\rho}(\bot) = \bot$.

Proof. Let $E_\bot$ be the set of equations imposing that all operations of Σ are strict, i.e., whenever an argument equals $\bot$ so does the result of the operation.

Next, we can equationally specify the function $\hat{\rho}$ as follows: for all constants of Σ,

$\hat{\rho}(\bot) = \bot$, $\hat{\rho}(c) = 0$

and for all functions f of Σ,

$\hat{\rho}(f(x_1, \ldots, x_n)) = \hat{\rho}(x_1) + (\hat{\rho}(x_2) + (\cdots + \hat{\rho}(x_n)..))$.

These together are equations $E(\hat{\rho})$.

Finally, each equation $e \equiv P = Q$ of E is replaced by $e_{\hat{\rho}} \equiv P + \hat{\rho}(Q) = Q + \hat{\rho}(P)$. These are the equations $E_{\hat{\rho}}$.

It is routine to prove that $(\Sigma_{\bot,\hat{\rho}}, E_\bot \cup E(\hat{\rho}) \cup E_{\hat{\rho}})$ is an initial algebra specification of $\mathsf{ADT}(A_\bot)$. □

The function $\hat{\rho}$ may be definable from the operations of the algebra A, as we will see in the case of the semigroup of natural numbers.

The *necessity* of an auxiliary function, such as $\hat{\rho}$, for lifting an algebraic specification from the abstract datatype of A to an algebraic specification for the abstract datatype of $A_\bot$ is not obvious. We will phrase this matter as an open question.

Problem 2.2. *Is there an example of a single sorted datatype A such that $\mathsf{ADT}(A)$ has a finite equational initial algebra specification while $\mathsf{ADT}(A_\bot)$ has no such specification.*

Closer to arithmetic datatypes, the following variation on the same question arises for semigroups:

Problem 2.3. *Is there an example of a single sorted datatype A which is equipped with a commutative and associative operation $\_ + \_$ and a constant 0 serving as a unit, such that $\mathsf{ADT}(A)$ has a finite equational initial algebra specification while $\mathsf{ADT}(A_\bot)$ has no such specification.*

Next, we note that $\bot$-enlargement yields no advantage from the perspective of algebraic specification of abstract datatypes.

Proposition 2.4. *Let Σ be a single sorted signature and assume that A is a Σ datatype with $\bot$-enlargement $A_\bot$. If $(\Sigma_\bot, E)$ is a finite equational initial algebra specification of $\mathsf{ADT}(A_\bot)$ then $\mathsf{ADT}(A)$ has a finite initial algebra specification as well.*

Proof. Let E' contain those equations of E in which $\bot$ does not occur on either side. Since A is a restriction (i.e., subalgebra of a reduct) of $A_\bot$, we have $A \models E'$. Now suppose that for closed Σ-terms t and r, $E \vdash t = r$ then the proof can only involve the application of equations from E which don't mention $\bot$ as otherwise both t and r must equal $\bot$ in $A_\bot$, which is not the case due to the construction of $A_\bot$ from A. So $E' \vdash t = r$. □

2.3 $\bot$-enlargement for partial algebras

The notion of a $\bot$-enlargement applied to total algebras does little useful work. However, it extends in a useful manner to partial algebras to make them total.

Suppose the application of a function f on arguments is undefined in a partial Σ-algebra A then we can take $\perp$ as the result of f on those arguments in $A_\perp$:

Definition 2.3. *Let A be a partial algebra with sort s such that A_s has at least two elements. Then (re)define the transformation $B = \mathsf{Enl}_\perp(A)$ to be a total algebra as follows: $B_s = A_s \cup \{\perp\}$ where $\perp$ is an absorptive element, and for $b_1, ..., b_n \in A_s$*

$$f(b_1, \ldots, b_n) = \perp \text{ in } B \text{ if, and only if, } f(b_1, \ldots, b_n) \text{ is undefined in } A.$$

Notice that earlier A was total or partial and $\mathsf{Enl}_\perp(A)$ simply added $\perp$ and ensured the operations respected it. Following Definition 2.3, in the case of A partial the notation $\mathsf{Enl}_\perp(A)$ implies that the partial operations of A have been made total using $\perp$.

Conversely, we can go the other way from a total algebra with absorbtive element $\perp$ to a partial algebra:

Definition 2.4. *Let A be an algebra with sort s and with $\perp$ as an absorptive element, and such that A_s has at least two elements. Then define the transformation $B = \mathsf{Pdt}_\perp(A)$ to be a partial algebra as follows: $B_s = A_s - \{\perp\}$ and for $b_1, ..., b_n, b \in B_s$,*

$$f(b_1, \ldots, b_n) = b \text{ in } B \text{ if, and only if, } f(b_1, \ldots, b_n) = b \text{ in } A.$$

Proposition 2.5. *Let A be a total algebra with sort s and with $\perp$ as an absorptive element, and such that A_s has at least two elements. Then*

$$\mathsf{Enl}_\perp(\mathsf{Pdt}_\perp(A)) = A.$$

Proposition 2.6. *Let A be an algebra with sort s. Then*

$$\mathsf{Pdt}_\perp(\mathsf{Enl}_\perp(A)) = A.$$

Thus, we can formulate a specification method for partial datatypes:

Definition 2.5. *Given a partial algebra A, construct $A_\perp$ and seek an initial algebra specification $(\Sigma_\perp, E)$ of the abstract datatype $\mathsf{ADT}(A_\perp)$. Then for all initial algebras $C \in I(\Sigma_\perp, E)$,*

$$A \cong \mathsf{Pdt}_\perp(C).$$

This $\perp$-enlargement approach to the specification of partial algebras makes use of conventional equational logic only, thereby avoiding the complications of various logics of partial functions. Arguably, $\perp$-enlargements provide a practical approach toward the design and analysis of such datatypes.

However, for total datatypes the situation is very different. From Proposition 2.4 above it follows that for the specification of datatypes which are total algebras the use of $\perp$-enlargements offers nothing new.

The following result formalises the fact $\perp$-enlargement is not of much use for classes of total algebras.

Theorem 2.2. *Let Σ be a single sorted signature. Let $K = Alg(\Sigma_\perp, E) \subseteq \mathsf{Enl}_\perp(Alg(\Sigma))$ (i.e., for all $A \in K$, $\mathsf{Pdt}_\perp(A)$ is a total algebra), and let E' consist of those equations of E in which $\perp$ does not occur on either side of the equation, then*

$$\bigcup_{A \in K} \mathsf{ADT}(\mathsf{Pdt}_\perp(A)) = Alg(\Sigma, E').$$

Proof. We write K' for $\bigcup_{A \in K} \mathsf{ADT}(\mathsf{Pdt}_\perp(A))$.

"$\subseteq$": Consider $B \in K'$ and equation $t = r \in E'$. Choose $A \in K$ so that $B \cong \mathsf{Pdt}_\perp(A)$. Then $A|_\Sigma \models t = r$ because $A \models t = r$ and $\perp$ does not feature in t and and not in r. Then $\mathsf{Pdt}_\perp(A) \models t = r$ because it is a substructure of $A|_\Sigma$, and $\mathsf{Pdt}_\perp(A) \models t = r$ because $B \cong \mathsf{Pdt}_\perp(A)$. This works for each equation in E' so that $B \in Alg(\Sigma, E')$.

"$\supseteq$": Let $B \in Alg(\Sigma, E')$, then $\mathsf{Enl}_\perp(B) \models t = r$. To see this notice that t and r must have the same free variables, otherwise $t = r$ cannot hold in any algebra in K. Now consider an equation $u = v$ in $E - E'$. If say u contains $\perp$ but v does not then $u = v$ cannot hold in any structure in $\mathsf{Enl}_\perp(Alg(\Sigma))$, as substituting values different from $\perp$ to all variables yields a counterexample. It follows that both u and v contain $\perp$ so that $\mathsf{Enl}_\perp(B) \models t = \perp = r$. We find that $\mathsf{Enl}_\perp(B) \in K$ so that $B = \mathsf{Pdt}_\perp(\mathsf{Enl}_\perp(B)) \in K'$. □

2.4 Many sorted case

Unfortunately, the definition of an absorptive element in the case of a many sorted signature is somewhat involved.

Definition 2.6. *Let A be a structure with many sorted signature Σ and let $s_1, \ldots, s_k \in \mathsf{sorts}(\Sigma)$. A Σ-structure A has* absorptive elements *for sorts $s_1, \ldots, s_k$ if the following conditions are met:*

(i) each function of Σ which takes at least one of its arguments in one of the sorts $s_1, \ldots, s_k$, yields a value in one of these sorts,

(ii) there are $a_1 \in A_{s_1}, \ldots, a_k \in A_{s_k}$, with the property that for each function f of Σ, if one of the arguments for f is in $\{a_1, \ldots, a_k\}$ then so is its value.

It is an open question whether the circumstances under which the family of absorptive elements is unique, allows an informative characterisation.

3 Teams

An additive semigroup is *cancellative* if it satisfies

$$x + y = x + z \text{ implies } y = z.$$

A cancellative additive semigroup may be equipped with a partial subtraction function as follows:

$$x - y = z \iff x = y + z.$$

$$(x+y)+z = x+(y+z) \tag{1}$$
$$x+y = y+x \tag{2}$$
$$x+0 = x \tag{3}$$
$$\rho(x) = x-x \tag{4}$$
$$(x+y)-y = x+\rho(y) \tag{5}$$
$$(x-y)+y = x+\rho(x-y) \tag{6}$$
$$x \neq \bot \to \rho(x) = 0 \tag{7}$$
$$x+\bot = \bot \tag{8}$$
$$\bot - x = \bot \tag{9}$$
$$x-\bot = \bot \tag{10}$$

Table 1: *Teams*: A set of axioms for teams

The cancellation property guarantees that subtraction thus defined is a single-valued relation.

Definition 3.1. *A* team *is a cancellative commutative additive semigroup equipped with a partial subtraction operator $x-y$ whose undefined values are $\bot$.*

Example 1. Consider a set $\mathbb{N}$ of natural numbers with these three additive structures:

$$N_{sg} = (\mathbb{N}|0,+)$$
$$N_{sg,-} = (\mathbb{N}|0,+,-)$$
$$N_{team} = (\mathbb{N}|0,+,-,\bot)$$

being the additive semigroup, the semigroup with partial subtraction and the team of natural numbers, respectively. The first two have the cancellation property. None of the stuctures are minimal.

A set *Teams* of first order axioms for teams is in Table 1.

The axioms of teams have various useful consequences. Taking $\rho(x) = x-x$ from axiom 4 the table we find:

1. (Non-triviality) $0 = \bot \to x = \bot$, i.e. if the team is nontrivial then $0 \neq \bot$. Proof: $x = x+0 = x+\bot = \bot$.
2. $\rho(\bot) = \bot$. Proof: By equation 5 on taking $x = 0$ and $y = \bot$.
3. $\rho(x+y) = \rho(x)+\rho(y)$. Proof: There are four cases according to $\rho(x) = 0$ and $\rho(y) = 0$ in each case the required fact follows immediately.
4. $\rho(0) = 0$. Proof: If $0 \neq \bot$ then axiom 7 yields $\rho(0) = 0$. Otherwise, if $0 = \bot$ then $\rho(0) = \rho(\bot) = \bot - \bot = \bot = 0$.

5. $x + y = \bot \to x = \bot \vee y = \bot$. Proof: If $0 = \bot$ then the team is trivial and the required fact holds trivially. So we assume $0 \neq \bot$ and moreover suppose that $x \neq \bot$ and $y \neq \bot$. It follows with axiom 7 that $\rho(x) = \rho(y) = 0$ so that $\rho(x + y) = 0 + 0 = 0$, whence, using $\rho(\bot) = \bot$, $x + y \neq \bot$.

6. (Cancellation) $x+y = x+z \to y+\rho(x) = z+\rho(x)$. Proof: If $x+y = x+z$ then $y + \rho(z) = (y + x) - x = (z + x) - x = z + \rho(x)$.

7. $(x + x) - x = x$. Proof: If $x = \bot$ the result is immediate, otherwise $\rho(x) = 0$ and $(x + x) - x = x + \rho(x) = x + 0 = x$.

8. $x - (y + z) = (x - y) - z$. Proof: We may assume that $0 \neq \bot, x \neq \bot, y \neq \bot$, and $z \neq \bot$, so that $\rho(x) = \rho(y) = \rho(z) = \rho(y + z) = 0$. Now: $(x - (y + z)) + (y + z) = x + \rho(y + z) = (x + \rho(y)) + \rho(z) = x$, and $((x - y) - z) + (y + z) = (((x - y) - z) + z) + y) = ((x - y) + \rho(z)) + y = (x - y) + y = x + \rho(y) = x$. The it follows with cancellation that $(x-(y+z))+\rho(y+z) = ((x-y)-z)+\rho(y+z)$ so that with $\rho(x+y) = 0$, $x - (y + z) = (x - y) - z$.

Note that ρ has the from of the auxiliary function in Proposition 2.3.
The following proposition asserts that $Teams$ axiomatises teams.

Proposition 3.1. *The non-trivial models of $Teams$ are the $\bot$-enlargements of cancellative commutative additive semigroups equipped with a partial subtraction function.*

Proof. Let H be a cancellative commutative additive semigroup with partial subtraction. It is immediate by inspection that the enlargement $H_\bot \models Teams$.

Conversely, let $K \models Teams$ be a nontrivial structure. The axioms of $Teams$ imply that $\bot$ is an absorptive element of K, so we can remove $\bot$ by $H = \mathsf{Pdt}_\bot(K)$.

The algebra H has signature $\Sigma_{sg,-} = (s, +, -, 0)$ and carrier $H_s = \{a \in K_s \mid a \neq \bot\}$ with $0 \in H_s$ as $0 \neq \bot$. By item 5 above, H_s, is closed under $+$ so that $+$ is total in H, and the first three axioms of $Teams$ guarantee that H is a commutative additive semigroup. Item 6 above implies that H is cancellative.

Now concerning subtraction: for $a, b, c \in H_s$, if $K \models a - b = c$ then $K \models c + b = (a - b) + b = a + \rho(a - b) = a$, which is precisely the criterion required for subtraction being defined in a cancellative semigroup. It follows that the $\bot$-enlargement K of H is a team (cf. Proposition 2.5). □

Proposition 3.2. *Teams cannot be defined by conditional equations, i.e., do not constitute a quasi-variety.*

Proof. Suppose a set E_{team} of conditional equations axiomatises the class of teams. Let c be a new constant then in all models of E_{team} for each interpretation of c: either $c = 0$ or $0 - c = \bot$ or $c + (0 - c) = 0$. It follows that at least one of these equations is derivable from E_{team} in which case E_{team} is not sound for all teams. □

We are unaware of an answer to the following question.

Problem 3.1. *Does the equational theory of teams have a finite basis (i.e. a finite equational axiomatisation)?*

Finally, because of our focus on subtraction $x - y$, we have defined teams for additive semigroups. The ideas apply to multiplicative semigroups and lead to a form of binary division x/y.

4 Splinters

A basic property of an algebra that acts as a datatype is that it is minimal. The algebras of the last section – in particular, the team of natural numbers – are not minimal. To remedy this, we define a splinter, which abstracts the idea of generating numbers from 0 using successor $s(x) = x + 1$.

4.1 Splinter property

Definition 4.1. *A single sorted algebra A of signature Σ with sort s is a* splinter *if there is a closed Σ-term t and a Σ-term $r(x)$ with (at most) one free variable x, so that their set,*

$$\{[\![t]\!], [\![r(t)]\!], [\![r(r(t))]\!], \dots\},$$

of values in A is the carrier of A.

The following is obvious:

Lemma 1. *Any Σ-algebra with the splinter property is Σ-minimal, i.e., a datatype.*

Example 2. First, note that the algebraic construction of the naturals from 0 with the successor function,

$$(\mathbb{N}|0, succ),$$

is a splinter: Σ contains a constant 0 and a unary function symbol S so that one may take $t \equiv 0$ and $r(x) \equiv S(x)$.

Recalling the algebras in Example 1, we add 1 as a constant to the additive semigroup, the semigroup with partial subtraction and the team of natural numbers, respectively:

$$N_{sg,1} = (\mathbb{N}|0, 1, +)$$
$$N_{sg,-,1} = (\mathbb{N}|0, 1, +, -)$$
$$N_{team,1} = (\mathbb{N}|0, 1, \bot, +, -)$$

These structures are splinters and so are minimal. In each case, the signature Σ contains 0 and a two place function $\_ + \_$ so that one can take $t \equiv 0$ and $r(x) \equiv x + 1$.

Adding an absorbtive element $\perp$ to a splinter requires special treatment:

Definition 4.2. *Let A be an algebra that is a splinter. Then $Enl_{\perp}(A)$ is a called a* common splinter*.*

On combining the constructions of (i) totalisation using teams and (ii) minimality using splinters, we have completed the first stage of our abstract analysis of the datatype of natural numbers with the partial operation of subtraction.

Splinters are of general use in the theory of algebraic specifications for computable abstract datatypes. A computable algebra A has a computable enumeration $\alpha : \mathbb{N} \to A$ – in which the operations and equality of A are computable; computability is an isomorphism invariant and so a property of abstract data types: see, for instance, [10].

Restricting to the single sorted case:

Proposition 4.1. *Each computable datatype has an enrichment to a splinter such that an equational initial algebra specification can be given for the corresponding abstract datatype.*

Proof. This can be found in the proofs of [10]. □

5 Natural numbers with multiplication

Adding multiplication to the additive semigroup of naturals, we obtain a semiring of natural numbers: $N_{sr} = (\mathbb{N}|0, 1, +, \cdot)$. This is a standard algebraic structure and is splinter.

5.1 Common semirings with subtraction

We will focus on the following algebra, which is a common semiring with subtraction, i.e., team extended by 1 and $\cdot$:

$$N_{team,1,\cdot} = (\mathbb{N}|0, 1, \perp, +, -, \cdot),$$

If Σ_{team} is the signature of teams then let $\Sigma_{team,\cdot,1}$ be the signature of $N_{team,1,\cdot}$. Recalling our notational conventions:

Definition 5.1. $\mathbf{N}_{team,1,\cdot}$ *is the isomorphism class of $N_{team,1,\cdot}$, i.e., the abstract datatype.*

Although we have defined the abstract datatype $\mathbf{N}_{team,1,\cdot}$ using a specific algebra $N_{team,1,\cdot}$, note that it is independent of the choice of parameters namely, the set $\mathbb{N}$ and the element $\perp$.

Without proof we mention:

Proposition 5.1. *The equations in Table 2 constitute an initial algebra specification of $\mathbf{N}_{team,1,\cdot}$.*

We notice that $N_{team,1,\cdot} \models (x+1)\cdot(y-z) = ((x+1)\cdot y) - ((x+1)\cdot z)$ while $N_{team,1,\cdot} \not\models x \cdot (y-z) = (x \cdot y) - (x \cdot z)$. Moreover, $N_{team,1,\cdot} \models x - x = 0 \cdot x$ while $N_{team,1,\cdot} \not\models 0 \cdot x = 0$.

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

$$x \cdot y = y \cdot x \tag{20}$$
$$x \cdot (y \cdot z) = (x \cdot y) \cdot z \tag{21}$$
$$1 \cdot x = x \tag{22}$$
$$(x+y) \cdot z = (x \cdot z) + (y \cdot z) \tag{23}$$
$$x \cdot \bot = \bot \tag{24}$$

Table 2: Specification of the abstract datatype $\mathbf{N}_{team,1,\cdot}$.

$$x-(y+z) = (x-y)-z \tag{25}$$
$$((x+x)-((1+1)+1))+((x+(x+x))-(((1+1)+1)+1)) = ((x+x)-((1+1)+1)) \tag{26}$$

$$(x+z)-(y+z) = (x-y)+(0 \cdot z) \tag{27}$$
$$(x-(y-z))+0 \cdot (x-y) = ((x-y)+z)+0 \cdot (y-z) \tag{28}$$
$$x-(y-z)) = ((x+z)-y)+0 \cdot (y-z) \tag{29}$$
$$(0 \cdot x)+(0 \cdot y) = 0 \cdot (x \cdot y) \tag{30}$$
$$(x+1) \cdot (y-z) = ((x+1) \cdot y)-((x+1) \cdot z) \tag{31}$$
$$(\underline{3} \cdot x)-\underline{4} = (\underline{5} \cdot x)-\underline{6} \tag{32}$$

Table 3: Some equations valid in $\mathbf{N}_{team,1,\cdot}$.

5.2 Equational theories: Impossibility of an ω-complete specification

An algebraic specification (Σ, E) with initial algebra $I(\Sigma, E)$ is called ω-*complete* or *inductively complete* if for all equations e over Σ,

$$I(\Sigma, E) \models e \iff Alg(\Sigma, E) \models e.$$

In other words, the equational theory of the initial model $Eqn(I(\Sigma, E))$ is identical to the equational theory of the whole specification $Eqn(Alg(\Sigma, E))$; equivalently, if all equations valid in its initial model are equationally derivable from it. A key consequence is this: The equational theory of $Eqn(I(\Sigma, E))$ has a finite basis if, and only if, it is ω-complete. See [21] for details.

Solvability of Diophantine equations over the integers is undecidable; see, e.g., the discussion in [24] of the work by Matijasevitch which finally solved Hilbert's 10-th problem in the negative. Solvability of diophantine equations $p(x_1, \ldots, x_k) = 0$ by non-negative integers is also undecidable: otherwise for an equation with n-variables, 2^n subcases can be constructed each of which deal with non-negative solutions only. So we find that *the non-solution of diophantine equations by non-negative integers is not computably enumerable.*

Proposition 5.2. *The equational theory of* $\mathbf{N}_{team,1,\cdot}$ *has no finite basis.*

Proof. If the equational theory of $\mathbf{N}_{team,1,\cdot}$ has a finite basis then the said equational theory is computably enumerable. Now, picking our concrete algebra, if $N_{team,1,\cdot} \models t = r$ were computably enumerable then so is the non-solution of diophantine equations over the natural numbers, which contradicts known facts. To see this reduction, consider a polynomial p with variables $x = x_1, \ldots, x_k$. We will reduce non-solvability of the equation $p = 0$ to validity of an equation in $N_{team,1,\cdot}$.

Given p write $p = q - r$ where q and r both contain $0, 1, +, \cdot$ only. Now $p = 0 \iff p^2 = 0 \iff (q^2 + r^2) - (2 \cdot q \cdot r) = 0$ so we find:

$$(\forall x \in \mathbb{N}^k)[p \neq 0] \iff N_{team,1,\cdot} \models 0 \cdot (((q \cdot q + r \cdot r) - (q \cdot r + q \cdot r)) - 1) = 0.$$

□

The last 6 equations in Table 3 are true in $N_{team,1,\cdot}$ and serve as an indication of the wealth of true equations in that setting.

6 Natural numbers without multiplication

Now consider the datatype $N_{team,1}$. This datatype is the reduct of $N_{team,1,\cdot}$ to the signature without multiplication, of course. The following basic fact is routine.

Proposition 6.1. *The first* 9 *equations of Table 2 constitute an initial algebra specification of* $\mathbf{N}_{team,1}$.

The first 2 equations of Table 3 are valid in $N_{team,1}$.

Proposition 6.2. *The equational theory of* $\mathbf{N}_{team,1}$ *is decidable.*

Proof. Validity of an equation over $N_{team,1}$ can be expressed as a first order sentence over Presburger arithmetic which is known to be decidable. We refer to [20] for information on Presburger arithmetic . □

Problem 6.1. *Does* $\mathbf{N}_{team,1}$ *have an* ω*-complete initial algebra specification (in other words: does its equational theory have a finite basis)?*

This problem has a simpler but yet open version as follows:

Problem 6.2. *Does the collection of equations involving a single variable only and valid in* $\mathbf{N}_{team,1}$ *have a finite axiomatisation by means of equations with any number of variables and valid in* $\mathbf{N}_{team,1}$*?*

7 Concluding remarks

7.1 Summary

In this note we have discussed some properties of arguably the most ubiquitous, or most obvious, example of a partial function: subtraction of natural numbers. Totalisation of subtraction in structures on the natural numbers is achieved by means of enlargement to a structure involving an absorptive element, here denoted $\perp$. Starting with the structure of an additive semigroup, where subtraction on the naturals belongs, we defined teams. Note that teams contain a binary subtraction operator, rather than a unary additive inverse operator as found in groups – remember an additive inverse does not have a role in the natural numbers. In principle, teams play a visible role in arithmetic as arithmetical datatypes for computation.

In exploring the fundamentals of arithmetical datatypes one encounters many distinct structures of interest even on the natural numbers. Having made definite choices of operations, the concepts, terminology and methods of abstract datatype theory become available for analysis, classification and new technical insights. They are also necessary: in pure mathematics, for example, the arithmetical structures are represented by now classical notions like semigroup, group, semiring, ring, field etc. But in practical arithmetical computations many more algebraic structures are needed, possibly designed in an *ad hoc* way for a specific purpose. In our studies of arithmetical structures these many algebras of interest need the explicit and uniform use of signatures, equations about operators, etc. These algebras may not appeal to pure mathematicians.

7.2 Totalisation and $\perp$

Unique absorptive elements play a key role in computer arithmetics, of which there are many. They have roles as flags for semantic behaviours, not only

partiality. Flags are commonly used to raise exceptions that need special attention; in arithmetical structures there are several. Inspired by floating point, in Anderson's transreal arithmetic [1, 15] the corresponding entity is referred to as *nullity*, which acts like a non-signalling NaN. Inspired by exact computer arithmetic based on intervals, in Setzer's wheels [28, 14] the absorptive element is denoted $\bot$, which is used to control undesired properties of ∞. We have studied transrationals and wheels of rationals using abstract datatype theory in [12] and [13], respectively.

In the common meadows of [9], the absorptive element is denoted with **a**, where it is used for totalising division. Here, and elsewhere, we have replaced **a** with $\bot$ but kept the term 'common' to indicate the presence of totalisation using an absorbtive element.

In terms of totalisation, division is an example of a partial operation on many classical arithmetical structures because of $1/0$; as a partial operation, division is second only to subtraction on natural numbers. Studies of division align with rings and fields, rather like subtraction aligns with semigroups and cancellative semigroups. A survey of options for totalisation of division can be found in [4].

Now, the authors of [25] insist that totalising division, should not lead to the incorporation of an absorptive element in the algebra of numbers, rather they claim that $1/0 = 0$ must be used. (We have not found written information on the point of view of the authors of [25] concerning subtraction, or in fact division, on naturals.) In the 1980s, logical aspects of algebra with $1/0 = 0$ have been first studied in a systematic manner in [26], however without any claim that so doing is mathematically necessary.

Pursuing the same $1/0 = 0$ approach in the setting of abstract datatypes is done in [11] and subsequently in [8] and [6], creating a theory of involutive meadows: if $0^{-1} = 0$ then inverse is an involution.

Turning to minimality, the term 'splinter' has been taken from [30]. A splinter with operations totalised by $\bot$ may be termed a 'common splinter', in accordance with [9] where a meadow equipped with an absorptive element, serving as the value of 0^{-1}, is referred to as a common meadow. Common meadows are to be contrasted with involutive meadows: if $0^{-1} = \bot$ then inverse is not an involution.

Admittedly, choosing $0 - 1 = 0$ is less problematic than adopting $1/0 = 0$, which runs counter to the intuition of most mathematicians. A conventional way to turn subtraction on naturals into a total function is to use a binary *monus* function that replaces $\bot$ by 0 and is denoted $\dot{-}$. Nevertheless, we hold that viewing $0 - 1$ as undefined, when working in non-negative integers is more plausible and sustainable. Adopting $\bot$ as a formalisation of "being undefined", so that $0 - 1 = \bot$, constitutes a straightforward formalisation of the intuition of "being undefined" in the context of all arithmetical datatypes.

This is not the place to go into the history of absorbtive elements and their uses. Our own introduction to them was in non-classical logics, especially the 3-valued logics of Kleene, designed to interpret the logic of computable predicates on the natural numbers [22]. Kleene's *weak 3-valued logic* has an absorbtive

element u of our kind (in his strong 3-valued logic u is not fully absorbtive for conjuction). Kleene's logics have a special place in the theory of non-classical logics [18]. The symbol $\perp$ is associated with the order-theoretic structures of denotational semantics where it is the lowest element 'bottom', e.g., [27]. Absorbtive elements for algebras and logics have been used in Hoare logics to reason about semantic errors in programming over abstract data types (e.g., [29]).

7.3 Subtraction

Expressions with subtraction as the leading function symbol deserve a name. We suggest to refer to $p - q$ loosely as "the difference of a and b"; and to call the expression $p - q$ a *diffterm* and its value a *diffsize*.

The contrast between differences and diffterms is similar to the contrast between fractions and ratios in [8]. The notations are often overloaded and obscured in expositions. Following the arguments of [17] regarding division, with notably fraction considered not to be a mathematical notion, it is then plausible that difference is not considered to be a mathematical notion either. In the case of addition, [5] argues for the use of "sumterm" for an expression of the form $p + q$, with sum standing for an ambiguous notion which depending on context may either be a value or an expression. We find that diffterms $5 - 2$ and $11 - 8$ are equivalent but not equal, whereas the diffsizes $[\![5 - 2]\!]$ and $[\![11 - 8]\!]$ are equal. The diffterm $p - q$ has p as its minuend and q as its subtrahend.

7.4 Perspectives

Arithmetical structures are designed to count, measure and, especially, compute. Rather than viewing algebras of natural numbers as classical abstract entities, both standardised and timeless, such numbers may be seen as having different roles and forms of existence. They are a classical object of study in mathematics, especially number theory and logic; they are used to model phenomena; they play a central role in numeracy and general education; they code texts, sounds and images; and they determine the scope and limits of practical computation and communication. In a more philosophical form, the idea of role is central to Benacerraf's commentary [3].

The role of computing challenges the classical algebra of arithmetic structures. The need for semantic flags create more structures. The natural numbers serve as a parameter type for lots of constructions of datatypes and abstract datatypes as discussed in detail in [5].

Emphasising the distinction between concrete and abstract datatypes, as we do, is not essential for a theory of abstract datatypes, which can be developed more abstractly without much mention of datatypes proper, and which may come close to the tenets of structuralism such as expounded in [2], for instance.

We think that maintaining the notion that a datatype implements an abstract datatype constitutes a fundamental intuition which ought to be introduced and preserved right from the start of the development of theories of

arithmetical datatypes. The latter view is compatible with our understanding of [19] where so-called practices enter the ontology of numbers. Datatypes may be understood as a way of capturing the (or an) essence of certain calculational practices.

References

[1] J.A. Anderson, N. Völker, and A. A. Adams. Perspecx Machine VIII, axioms of transreal arithmetic. In J. Latecki, D. M. Mount and A. Y. Wu (editors) *Vision Geometry XV*, (2007). https://doi.org/10.1117/12.698153 .

[2] Steve Awodey. Structuralism, invariance, and univalence. *Philosophia Mathematica* 22 (1), 1-11, (2014). https://doi.org/10.1093/philmat/nkt030.

[3] Paul Benacerraf. What numbers could not be. *The Philosophical Review* 74 (1), 47–73, (1965).

[4] J.A. Bergstra. Division by zero, a survey of options. *Transmathematica 2019*, (published 2019-06-25), https://doi.org/10.36285/tm.v0i0.17, (2019).

[5] J.A. Bergstra. Sumterms, summands, sumtuples, and sums and the meta-arithmetic of summation. *Scientific Annals of Computer Science* 30 (2), 167–203, (2020).

[6] J.A. Bergstra, I. Bethke. Subvarieties of the variety of meadows. *Scientific Annals of Computer Science.* 27 (1), 1–18, (2017).

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

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

[9] J.A. Bergstra and A. Ponse. 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, 46-61 (2015). Improved version: https://arxiv.org/pdf/1406.6878v4.pdf.

[10] J.A. Bergstra and J.V. Tucker. Equational specifications, complete term rewriting systems, and computable and semi-computable algebras. *Journal of the ACM*, 42 (6), 1194–1230, (1995).

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

[12] J.A. Bergstra and J.V. Tucker. Transrational numbers as an abstract data type. *Transmathematica 2020*, (published 2020-12-16), https://doi.org/10.36285/tm.v0i0.17, (2020).

[13] J.A. Bergstra and J.V. Tucker. The wheel of rational numbers as an abstract data type. In P. James and M. Roggenbach (editors), *Workshop on Algebraic Development Techniques*, Lecture Notes in Computer Science 12669, 1-18, (2021).

[14] J. Carlström. Wheels – On division by zero. *Mathematical Structures in Computer Science*, 14 (1), 143–184, (2004).

[15] T.S. dos Reis, W. Gomide, and J.A.D.W. Anderson. Construction of the transreal numbers and algebraic transfields. *IAENG International Journal of Applied Mathematics*, 46 (1), 11–23, (2016). http://www.iaeng.org/IJAM/issues_v46/issue_1/IJAM_46_1_03.pdf

[16] H. D. Ehrich, M. Wolf, and J. Loeckx. *Specification of Abstract Data Types.* Vieweg & Teubner, (1997).

[17] Martha I.F. Fandino Pinilla. Fractions: conceptual and didactic aspects. *Acta Didactica Universitatis Comenianae*, 7, 82–115, (2007).

[18] Kleene's three valued logics and their children. *Fundamenta Informaticae*, 20, (1-3), 113-131, (1994).

[19] J. D. Godino et al. Why is the learning of elementary arithmetic concepts difficult? Semiotic tools for understanding the nature of mathematical objects. *Educational Studies in Mathematics*, 77, 247–265 (2011). https://doi.org/10.1007/s10649-010-9278-x

[20] Christoph Haase. A survival guide to Presburger arithmetic. *ACM SIGLOG News*, 5 (3), 67–81, (2018).

[21] J. Heering. Partial evaluation and the ω-completeness of algebraic specifications. *Theoretical Computer Science*, 43, 149-167, (1986).

[22] S. C. Kleene. *Introduction to metamathematics.* North-Holland Pub. Co., Amsterdam (1952).

[23] Susan J. Lamon. *Teaching fractions and ratios for understanding – Essential content knowledge and instructional strategies for teachers.* Routledge, 1999. (Third Edition 2012.)

[24] Y. Manin. *A course in mathematical logic.* Springer, (1977). (Second Edition 2010.)

[25] H. Michiwaki, S. Saitoh, and N. Yamada. Reality of the division by zero $z/0 = 0$. *International Journal of Applied Physics and Mathematics*, 6, 1–8, (2016). http://dx.doi.org/10.17706/ijapm.2016.6.1.1-8

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

[27] D. S. Scott. Data types as lattices. *SIAM Journal on Computing*, 5, 522–587, (1976).

[28] A. Setzer. Wheels, Unpublished draft, (1997). http://www.cs.swan.ac.uk/~csetzer/articles/wheel.pdf.

[29] J. V. Tucker and J. I. Zucker. *Program correctness over abstract data types with error-state semantics.* North-Holland Pub. Co., Amsterdam (1988).

[30] J. S. Ullian. Splinters of recursive functions. *Journal of Symbolic Logic*, 25 (1) 33–38, (1960).