University of Amsterdam

Programming Research Group

Inversive Meadows and Divisive Meadows

J.A. Bergstra
C.A. Middelburg

Report PRG0907 July 2009

J.A. Bergstra

Programming Research Group
Faculty of Science
University of Amsterdam

Kruislaan 403
1098 SJ   Amsterdam
The Netherlands

tel. +31 20 525.7591
e-mail: janb@science.uva.nl

C.A. Middelburg

Programming Research Group
Faculty of Science
University of Amsterdam

Kruislaan 403
1098 SJ   Amsterdam
The Netherlands

e-mail: kmiddelb@science.uva.nl

Programming Research Group Electronic Report Series

Inversive Meadows and Divisive Meadows

J.A. Bergstra and C.A. Middelburg

Informatics Institute, Faculty of Science, University of Amsterdam,
Science Park 107, 1098 XG Amsterdam, the Netherlands
J.A.Bergstra@uva.nl,C.A.Middelburg@uva.nl

Abstract. An inversive meadow is a commutative ring with identity and a total multiplicative inverse operation satisfying $0^{-1} = 0$. Previously, inversive meadows were shortly called meadows. In this paper, we introduce divisive meadows, which are inversive meadows with the multiplicative inverse operation replaced by a division operation. We introduce a translation from the terms over the signature of divisive meadows into the terms over the signature of inversive meadows and a translation the other way round to show that it depends on the angle from which they are viewed whether inversive meadows or divisive meadows must be considered more basic. Divisive meadows are more basic if variants with a partial multiplicative inverse or division operation are considered as well. We also take a survey of first-order logics that are appropriate to handle those partial variants of inversive and divisive meadows.

Keywords: inversive meadow, divisive meadow, projection semantics, partial algebra construction, first-order logic.

MSC2000 codes: 12E12, 12E30, 12L12.

1 Introduction

The primary mathematical structure for measurement and computation is unquestionably a field. In [10], meadows are proposed as alternatives for fields with a purely equational specification. A meadow is a commutative ring with identity and a total multiplicative inverse operation satisfying two equations which imply that the multiplicative inverse of zero is zero. Thus, meadows are total algebras. As usual in field theory, the convention to consider $p \mathbin{/} q$ as an abbreviation for $p \cdot (q^{-1})$ was used in subsequent work on meadows (see e.g. [5, 8]). This convention is no longer satisfactory if partial variants of meadows are considered too, as will be demonstrated in this paper. That is why we rename meadows into inversive meadows and introduce divisive meadows. A divisive meadow is an inversive meadow with the multiplicative inverse operation replaced by the division operation suggested by the above-mentioned abbreviation convention. Henceforth, we will use the name meadow whenever the distinction between inverse meadows and divisive meadows is not important.

For both inversive meadows and divisive meadows, we give an equational specification. Given the equational specification of inversive meadows, we can easily give a modular specification of divisive meadows using module algebra [4].

We give the modular specification in question and show that the equational theory associated with it is the same as the equational theory associated with the equational specification of divisive meadows.

Partial variants of meadows are obtained by turning the total multiplicative inverse or division operation into a partial one. There is one way in which the multiplicative inverse operation can be turned into a partial operation, whereas there are two conceivable ways in which the division operation can be turned into a partial operation. In [7], projection semantics is proposed as an approach to define the meaning of programs. Projection semantics explains the meaning of programs in terms of known programs instead of in terms of more or less sophisticated mathematical objects. We transpose this approach to the current setting so as to explain the meaning of terms over the signature of divisive meadows in terms of terms over the signature of inversive meadows. It is also used to explain the meaning of terms over the signature of inversive meadows in terms of terms over the signature of divisive meadows. It happens that the latter projection is appropriate in the case of the partial variants of inversive meadows as well, whereas the former projection is not appropriate in the case of one of the kinds of partial variants of divisive meadows. That is why we take divisive meadows for more basic than inversive meadows if their partial variants are considered as well.

The main inversive meadow that we are interested in is $\mathcal{Q}_0^{\mathrm{i}}$, the zero-totalized field of rational numbers, which differs from the field of rational numbers only in that the multiplicative inverse of zero is zero. The main divisive meadow that we are interested in is $\mathcal{Q}_0^{\mathrm{d}}$, which is $\mathcal{Q}_0^{\mathrm{i}}$ with the multiplicative inverse operation replaced by a division operation in conformity with the first projection referred to above. We give finite equational specifications of which $\mathcal{Q}_0^{\mathrm{i}}$ and $\mathcal{Q}_0^{\mathrm{d}}$ are the initial algebras. We obtain a partial variant from $\mathcal{Q}_0^{\mathrm{i}}$ and two partial variants from $\mathcal{Q}_0^{\mathrm{d}}$ by means of a rather unknown, but simple construction. This fits in with our position that partial algebras should be made of total ones. Thus, we have five algebras related to rational numbers, each requiring only equational logic for total algebras as a tool for their construction.

Having constructed the partial variant of $\mathcal{Q}_0^{\mathrm{i}}$, the question whether it satisfies the equation $0^{-1} = 0^{-1}$ and related questions are still open because the logic of partial functions to be used when working with it has not been fixed yet. Similar remarks apply to the partial variants of $\mathcal{Q}_0^{\mathrm{d}}$. This means that it is still a matter of design which logic of partial functions can be used for reasoning. Therefore, we take a survey of first-order logics that may be appropriate to handle the partial variants of $\mathcal{Q}_0^{\mathrm{i}}$ and $\mathcal{Q}_0^{\mathrm{d}}$.

It appears that, in the sphere of groups, rings and fields, the qualifications "inversive" and "divisive" have only been used by Yamada [34] and Verloren van Themaat [32], respectively. Our use of these qualifications is in line with theirs.

This paper is organized as follows. First, we give a brief summary of inverse meadows (Section 2). Next, we introduce divisive meadows and relate them to inverse meadows by means of projection semantics (Section 3). After that, we use module algebra to give a modular specification that corresponds to the equational specification of divisive meadows given earlier (Section 4). Then, we introduce simple constructions of partial inversive and divisive meadows from total ones (Section 5). Following this, we introduce the inversive meadow of rational numbers, the divisive meadow of rational numbers, and partial variants of them (Section 6). After that, we take surveys of two-valued and three-valued first-order logics that may be appropriate to handle partial algebras (Sections 7 and 8). Following this, we discuss issues on which the suitability of such a logic for a particular purpose may depend (Section 9). Finally, we make some concluding remarks (Section 10).

2 Inversive Meadows

In this section, we give a brief summary of inversive meadows. In [10], inversive meadows were introduced for the first time. They are further investigated in e.g. [5, 8, 11].

An inversive meadow is a commutative ring with identity and a total multiplicative inverse operation satisfying two equations which imply that the multiplicative inverse of zero is zero.

The signature of inversive meadows consists of the following constants and operators:

- the constants 0 and 1;
- the binary *addition* operator $+$;
- the binary *multiplication* operator $\cdot$;
- the unary *additive inverse* operator $-$;
- the unary *multiplicative inverse* operator ${}^{-1}$.

We assume that there are infinitely many variables, including x, y and z. Terms are build as usual. We use infix notation for the binary operators, prefix notation for the unary operator $-$, and postfix notation for the unary operator ${}^{-1}$. Moreover, we use the usual precedence convention to reduce the need for parentheses. We introduce subtraction as an abbreviation: $p - q$ abbreviates $p+(-q)$. For each natural number n, we write $\underline{n}$ for the numeral for n. That is, the term $\underline{n}$ is defined by induction on n as follows: $\underline{0} = 0$ and $\underline{n+1} = \underline{n} + 1$. We also use the notation p^n for exponentiation with a natural number as exponent. For each term p over the signature of meadows, the term p^n is defined by induction on n as follows: $p^0 = 1$ and $p^{n+1} = p^n \cdot p$.

The constants and operators from the signature of inversive meadows are adopted from rational arithmetic, which gives an appropriate intuition about these constants and operators. The set of all terms over the signature of inversive meadows constitutes the *inversive meadow notation*.

An inversive meadow is an algebra over the signature of inversive meadows that satisfies the equations given in Tables 1 and 2. The equations given in Table 1 are the axioms of a commutative ring with identity. Thus, an inversive meadow is a commutative ring with identity and a total multiplicative inverse operation ${}^{-1}$ satisfying the reflexivity equation $(x^{-1})^{-1} = x$ and the restricted

Table 1. Axioms of a commutative ring with identity

$(x + y) + z = x + (y + z)$	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$
$x + y = y + x$	$x \cdot y = y \cdot x$
$x + 0 = x$	$x \cdot 1 = x$
$x + (-x) = 0$	$x \cdot (y + z) = x \cdot y + x \cdot z$

Table 2. Additional axioms for an inversive meadow

$(x^{-1})^{-1} = x$
$x \cdot (x \cdot x^{-1}) = x$

inverse equation $x \cdot (x \cdot x^{-1}) = x$. From the equations given in Tables 1 and 2, the equation $0^{-1} = 0$ is derivable.

The advantage of working with a total multiplicative inverse operation lies in the fact that conditions like $x \neq 0$ in $x \neq 0 \Rightarrow x \cdot x^{-1} = 1$ are not needed to guarantee meaning.

A *non-trivial inversive meadow* is an inversive meadow that satisfies the *separation axiom* $0 \neq 1$; and an *inversive cancellation meadow* is an inversive meadow that satisfies the *cancellation axiom* $x \neq 0 \wedge x \cdot y = x \cdot z \Rightarrow y = z$, or equivalently, the *general inverse law* $x \neq 0 \Rightarrow x \cdot x^{-1} = 1$. An important property of non-trivial inversive cancellation meadows is the following: $0 \cdot (0^{-1}) = 0$, whereas $x \cdot (x^{-1}) = 1$ for $x \neq 0$.

Henceforth, we will write Σ_{imd} for the signature of inversive meadows and E_{imd} for the set of axioms for inversive meadows.

3 Divisive Meadows

In this section, we introduce divisive meadows and use projection semantics to explain the terms over the signature of divisive meadows in terms of terms over the signature of inversive meadows.

A divisive meadow is a commutative ring with identity and a total division operation satisfying three equations which imply that division by zero always yields zero.

The signature of divisive meadows is the signature of inversive meadows with the unary multiplicative inverse operator $^{-1}$ replaced by:

– the binary *division* operator $/$.

The set of all terms over the signature of divisive meadows constitutes the *divisive meadow notation.*

A divisive meadow is an algebra over the signature of divisive meadows that satisfies the equations given in Tables 1 and 3. Thus, a divisive meadow is a commutative ring with identity and a total division operation $/$ satisfying the three equations given in Table 3. The first two of these equations are the obvious

Table 3. Additional axioms for a divisive meadow

$1 \mathbin{/} (1 \mathbin{/} x) = x$
$(x \cdot x) \mathbin{/} x = x$
$x \mathbin{/} y = x \cdot (1 \mathbin{/} y)$

counterparts of the additional axioms for inversive meadows. From the equations given in Tables 1 and 3, the equation $x \mathbin{/} 0 = 0$ is derivable. The equation $1 \mathbin{/} 0 = 0$ can be derived without using the last equation from Table 3, and then the latter equation can be applied to derive the equation $x \mathbin{/} 0 = 0$.

A *non-trivial divisive meadow* is an divisive meadow that satisfies the *separation axiom*; and a *divisive cancellation meadow* is an divisive meadow that satisfies the *cancellation axiom*. An important property of non-trivial divisive cancellation meadows is the following: $0 \mathbin{/} 0 = 0$, whereas $x \mathbin{/} x = 1$ for $x \neq 0$.

Henceforth, we will write Σ_{dmd} for the signature of divisive meadows and E_{dmd} for the set of axioms for divisive meadows.

We can explain the meaning of the terms over the signature of divisive meadows by means of a projection `dmn2imn` from the divisive meadow notation to the inversive meadow notation. This projection is defined as follows:

$$
\begin{aligned}
&\mathtt{dmn2imn}(x) = x \;, \\
&\mathtt{dmn2imn}(0) = 0 \;, \\
&\mathtt{dmn2imn}(1) = 1 \;, \\
&\mathtt{dmn2imn}(p + q) = \mathtt{dmn2imn}(p) + \mathtt{dmn2imn}(q) \;, \\
&\mathtt{dmn2imn}(p \cdot q) = \mathtt{dmn2imn}(p) \cdot \mathtt{dmn2imn}(q) \;, \\
&\mathtt{dmn2imn}(-p) = -\mathtt{dmn2imn}(p) \;, \\
&\mathtt{dmn2imn}(p \mathbin{/} q) = \mathtt{dmn2imn}(p) \cdot (\mathtt{dmn2imn}(q)^{-1}) \;.
\end{aligned}
$$

The projection `dmn2imn` supports an interpretation of the theory of divisive meadows in the theory of inversive meadows: for each equation $p = q$ derivable from the axioms of a divisive meadow, the equation $\mathtt{dmn2imn}(p) = \mathtt{dmn2imn}(q)$ is derivable from the axioms of a inversive meadow.[1] Therefore the projection `dmn2imn` determines a mapping from divisive meadows to inversive meadows.

We can also explain the meaning of the terms over the signature of inversive meadows by means of a projection `imn2dmn` from the inversive meadow notation to the divisive meadow notation. This projection is defined as follows:

$$
\begin{aligned}
&\mathtt{imn2dmn}(x) = x \;, \\
&\mathtt{imn2dmn}(0) = 0 \;, \\
&\mathtt{imn2dmn}(1) = 1 \;, \\
&\mathtt{imn2dmn}(p + q) = \mathtt{imn2dmn}(p) + \mathtt{imn2dmn}(q) \;, \\
&\mathtt{imn2dmn}(p \cdot q) = \mathtt{imn2dmn}(p) \cdot \mathtt{imn2dmn}(q) \;, \\
&\mathtt{imn2dmn}(-p) = -\mathtt{imn2dmn}(p) \;, \\
&\mathtt{imn2dmn}(p^{-1}) = 1 \mathbin{/} \mathtt{imn2dmn}(p) \;.
\end{aligned}
$$

[1] For the notion of a translation that supports a theory interpretation, see e.g. [33].

The projection `imn2dmn` supports an interpretation of the theory of inversive meadows in the theory of divisive meadows: for each equation $p = q$ derivable from the axioms of a inversive meadow, the equation $\mathtt{imn2dmn}(p) = \mathtt{imn2dmn}(q)$ is derivable from the axioms of a divisive meadow. Therefore the projection `imn2dmn` determines a mapping from inversive meadows to divisive meadows.

4 Modular Specification of Divisive Meadows

In this section, we give a modular specification of divisive meadows using basic module algebra [4].

BMA[*fol*] (Basic Module Algebra for *f*irst-*o*rder *l*ogic specifications) is a many-sorted equational theory of modules which covers the concepts on which the key modularization mechanisms found in existing specification formalisms are based. The signature of *BMA*[*fol*] includes among other things:

- the sorts *ATSIG* of *atomic signatures*, *ATREN* of *atomic renamings*, *SIG* of *signatures*, and *M* of *modules*;
- the binary *deletion* operator $\Delta : \mathit{ATSIG} \times \mathit{SIG} \to \mathit{SIG}$;
- the unary *signature* operator $\Sigma : M \to \mathit{SIG}$;
- for each first-order sentence ϕ over some signature, the constant $\langle \phi \rangle : M$;
- the binary *renaming application* operator $\,. : \mathit{ATREN} \times M \to M$;
- the binary *combination* operator $+ : M \times M \to M$;
- the binary *export* operator $\square : \mathit{SIG} \times M \to M$.

The axioms of *BMA*[*fol*] as well as four different models for *BMA*[*fol*] can be found in [4]. A useful derived operator is the *hiding* operator $\Delta : \mathit{ATSIG} \times M \to M$ defined by $a \Delta X = (a \Delta \Sigma(X)) \square X$. Below, we will use the notational conventions introduced in Section 3.5 of [4].

Let Md_i be the closed module expression corresponding to the equational specification of inversive meadows, i.e. $\langle (x + y) + z = x + (y + z) \rangle + \cdots + \langle x \cdot (x \cdot x^{-1}) = x \rangle$. We give a modular specification of divisive meadows using *BMA*[*fol*] as follows:

$$\mathit{Md}_d = \mathbf{F} : {}^{-1} : Q \to Q \;\Delta\; (\mathit{Md}_i + \langle x \,/\, y = x \cdot (y^{-1}) \rangle) \;.$$

In [4], a semantic mapping *EqTh* is defined that gives, for each closed module expression, its equational theory. We have the following theorem:

Theorem 1. *$\mathit{EqTh}(\mathit{Md}_d)$ is the equational theory associated with the equational specification of divisive meadows given in Section 3.*

Proof. In [4], a semantic mapping *Mod* is defined that gives, for each closed module expression, its model class. *Mod* and *EqTh* are defined such that $\mathit{EqTh}(m)$ is the equational theory of $\mathit{Mod}(m)$ for each closed module expression m. Hence, it is sufficient to show that $\mathit{Mod}(\mathit{Md}_d)$ is the class of models of the equational specification of divisive meadows. By the definition of *Mod*, we have to show that: (i) the reduct to the signature of divisive meadows of each model of the equational specification of inversive meadows extended with the equation $x \,/\, y = x \cdot (y^{-1})$ is a model of the equational specification of divisive meadows; (ii) each model of the equational specification of divisive meadows can be expanded with a multiplicative inverse operation satisfying $(x^{-1})^{-1} = x$ and $x \cdot (x \cdot x^{-1}) = x$. Using the equations from the equational specification of inversive meadows and the equation $x \,/\, y = x \cdot (y^{-1})$, it can easily be proved by equational reasoning that all equations from the equational specification of divisive meadows are satisfied by the reducts in question. Let ${}^{-1}$ be defined by $x^{-1} = 1 \,/\, x$. Then, using the equations from the equational specification of divisive meadows and the equation $x^{-1} = 1 \,/\, x$, it can easily be proved by equational reasoning that the equations $(x^{-1})^{-1} = x$ and $x \cdot (x \cdot x^{-1}) = x$ are satisfied by the expansions in question. $\Box$

We give the following modular specification of *reduced divisive meadows*:

$$
\begin{aligned}
Md_{rd1} &= \mathbf{F} : \cdot : Q \times Q \to Q \,\Delta\, Md_{d}\ ,\\
Md_{rd2} &= \mathbf{F} : - : Q \to Q \,\Delta\, (Md_{rd1} + \langle x - y = x + (-y)\rangle)\ ,\\
Md_{rd3} &= \mathbf{F} : + : Q \times Q \to Q \,\Delta\, Md_{rd2}\ ,\\
Md_{rd} &= \mathbf{F} : 0 : Q \,\Delta\, Md_{rd3}\ .
\end{aligned}
$$

The signature of reduced divisive meadows consists of the constant 1 and the binary operators $-$ and $/$. We can explain the meaning of the terms over the signature of inversive meadows by means of a projection `imn2rdmn` to terms over the signature of reduced divisive meadows. This projection is defined as follows:

$$
\begin{aligned}
&\texttt{imn2rdmn}(x) = x\ ,\\
&\texttt{imn2rdmn}(0) = 1 - 1\ ,\\
&\texttt{imn2rdmn}(1) = 1\ ,\\
&\texttt{imn2rdmn}(p + q) = \texttt{imn2rdmn}(p) - ((1 - 1) - \texttt{imn2rdmn}(q))\ ,\\
&\texttt{imn2rdmn}(p \cdot q) = \texttt{imn2rdmn}(p) \,/\, (1 \,/\, \texttt{imn2rdmn}(q))\ ,\\
&\texttt{imn2rdmn}(-p) = (1 - 1) - \texttt{imn2rdmn}(p)\ ,\\
&\texttt{imn2rdmn}(p^{-1}) = 1 \,/\, \texttt{imn2rdmn}(p)\ .
\end{aligned}
$$

We have the following theorem:

Theorem 2. *$EqTh(Md_{rd})$ is the equational theory associated with the equational specification whose equations are given in Table 4.*

Proof. The proof follows the same line as the proof of Theorem 1. For the expansion, we define zero, addition, multiplication, and additive inverse as follows: $0 = 1 - 1$, $x + y = x - ((1 - 1) - y)$, $x \cdot y = x \,/\, (1 \,/\, y)$, and $-x = (1 - 1) - x$. $\Box$

Using the equational specification of reduced divisive meadows, it is easy to show that the projection `imn2rdmn` supports an interpretation of the theory of inversive meadows in the theory of reduced divisive meadows.

The following are some open problems concerning inversive meadows, divisive meadows, and reduced divisive meadows:

Table 4. Axioms of a reduced divisive meadow

$(x - ((1 - 1) - y)) - ((1 - 1) - z) = x - ((1 - 1) - (y - ((1 - 1) - z)))$
$x - ((1 - 1) - y) = y - ((1 - 1) - x)$
$x - (1 - 1) = x$
$x - x = 1 - 1$
$(x \,/\, (1 \,/\, y)) \,/\, (1 \,/\, z) = x \,/\, (1 \,/\, (y \,/\, (1 \,/\, z)))$
$x \,/\, (1 \,/\, y) = y \,/\, (1 \,/\, x)$
$x \,/\, 1 = x$
$x \,/\, (1 \,/\, (y - ((1 - 1) - z))) = x \,/\, (1 \,/\, y) - ((1 - 1) - (x \,/\, (1 \,/\, z)))$
$(x \,/\, (1 \,/\, x)) \,/\, x = x$

- do there exist equational specifications of inversive meadows, divisive meadows, and reduced divisive meadows with less than 10 equations, 11 equations, and 9 equations, respectively;
- can the number of binary operators needed to explain the meaning of the terms over the signature of inversive meadows be reduced to one.

5 Partial Inversive and Divisive Meadows

In this section, we introduce simple constructions of partial inversive and divisive meadows from total ones and show why we take divisive meadows for more basic than inversive meadows if the partial ones are considered as well.

We take the position that partial algebras should be made from total ones. For the particular case of meadows, this implies that relevant partial meadows are obtained by making operations undefined for certain arguments.

Let $\mathcal{M}_i$ be an inversive meadow. Then it make sense to construct one partial inversive meadow from $\mathcal{M}_i$:

- $0^{-1} \uparrow \mathcal{M}_i$ is the partial algebra that is obtained from $\mathcal{M}_i$ by making 0^{-1} undefined.

Let $\mathcal{M}_d$ be a divisive meadow. Then it make sense to construct two partial divisive meadows from $\mathcal{M}_d$:

- $Q \,/\, 0 \uparrow \mathcal{M}_d$ is the partial algebra that is obtained from $\mathcal{M}_d$ by making $q \,/\, 0$ undefined for all q in the domain of $\mathcal{M}_d$;
- $(Q \setminus \{0\}) \,/\, 0 \uparrow \mathcal{M}_d$ is the partial algebra that is obtained from $\mathcal{M}_d$ by making $q \,/\, 0$ undefined for all q in the domain of $\mathcal{M}_d$ different from 0.

Clearly, the partial meadow constructions are special cases of a more general partial algebra construction for which we have coined the term *punching*. Presenting the details of the general construction is outside the scope of the current paper.

Let $\mathcal{M}_i$ be an inversive meadow and let $\mathcal{M}_d$ be an divisive meadow. It happens that the projection imn2dmn recovers $0^{-1} \uparrow \mathcal{M}_i$ from $Q \,/\, 0 \uparrow \mathcal{M}_d$ as well as $(Q \setminus \{0\}) \,/\, 0 \uparrow \mathcal{M}_d$, the projection dmn2imn recovers $Q \,/\, 0 \uparrow \mathcal{M}_d$ from $0^{-1} \uparrow \mathcal{M}_i$, and the projection dmn2imn does not recover $(Q \setminus \{0\}) \,/\, 0 \uparrow \mathcal{M}_d$ from $0^{-1} \uparrow \mathcal{M}_i$:

- 0^{-1} is undefined in $0^{-1} \uparrow \mathcal{M}_i$, $\texttt{imn2dmn}(0^{-1}) = 1 \,/\, 0$, and $1 \,/\, 0$ is undefined in $Q \,/\, 0 \uparrow \mathcal{M}_d$ and $(Q \setminus \{0\}) \,/\, 0 \uparrow \mathcal{M}_d$;
- $x \,/\, 0$ is undefined in $Q \,/\, 0 \uparrow \mathcal{M}_d$, $\texttt{dmn2imn}(x \,/\, 0) = x \cdot (0^{-1})$, and $x \cdot (0^{-1})$ is undefined in $0^{-1} \uparrow \mathcal{M}_i$;
- $0 \,/\, 0 = 0$ in $(Q \setminus \{0\}) \,/\, 0 \uparrow \mathcal{M}_d$, $\texttt{dmn2imn}(0 \,/\, 0) = 0 \cdot (0^{-1})$, but $0 \cdot (0^{-1})$ is undefined in $0^{-1} \uparrow \mathcal{M}_i$.

This uncovers that $(Q \setminus \{0\}) \,/\, 0 \uparrow \mathcal{M}_d$ expresses a view on the partiality of division by zero that cannot be expressed if only multiplicative inverse is available. Therefore, we take divisive meadows for more basic than inversive meadows if their partial variants are considered as well. Otherwise, we might take inversive meadows for more basic, e.g. because of supposed notational simplicity. Thus, the move from a total algebra to a partial algebra may imply a reversal of the preferred direction of projection from dmn2imn to imn2dmn. This shows that projection semantics is a tool within a setting: if the setting changes, the tool, or rather its way of application, changes as well.

Returning to $(Q \setminus \{0\}) \,/\, 0 \uparrow \mathcal{M}_d$, the question remains whether the equation $0 \,/\, 0 = 0$ is natural. The total cost C_n of producing n items of some product is often viewed as the sum of a fixed cost FC and a variable cost VC_n. Moreover, for $n \geq 1$, the variable cost VC_n of producing n items is usually viewed as n times the marginal cost per item, taking $VC_n \,/\, n$ as the marginal cost per item. For $n = 0$, the variable cost of producing n items and the marginal cost per item are both 0. This makes the equation $VC_0 \,/\, 0 = 0$ natural.

6 Algebras Related to Rational Numbers

In this section, we obtain first an inverse meadow and a divisive meadow closely related to the field of rational numbers as the initial algebras of equational specifications and next partial variants of those meadows as the result of the partial algebra construction introduced in Section 5. As usual, we write $I(\Sigma, E)$ for the initial algebra among the algebras over the signature Σ that satisfy the equations E (see e.g. [9]).

The inversive meadow that we are interested in is $\mathcal{Q}_0^{\mathrm{i}}$, the inversive meadow of rational numbers:

$$\mathcal{Q}_0^{\mathrm{i}} = I(\Sigma_{\mathrm{imd}}, E_{\mathrm{imd}} \cup \{(1 + x^2 + y^2) \cdot (1 + x^2 + y^2)^{-1} = 1\}) \; .$$

At the end of this section, we will prove that $\mathcal{Q}_0^{\mathrm{i}}$ differs from the field of rational numbers only in that the multiplicative inverse of zero is zero. The divisive meadow that we are interested in is $\mathcal{Q}_0^{\mathrm{d}}$, the divisive meadow of rational numbers:

$$\mathcal{Q}_0^{\mathrm{d}} = I(\Sigma_{\mathrm{dmd}}, E_{\mathrm{dmd}} \cup \{(1 + x^2 + y^2) \,/\, (1 + x^2 + y^2) = 1\}) \; .$$

$\mathcal{Q}_0^{\mathrm{d}}$ differs from $\mathcal{Q}_0^{\mathrm{i}}$ only in that the multiplicative inverse operation replaced by a division operation in conformity with the projection `imn2dmn` defined in Section 3.

We are also interested in the partial meadows that can be obtained from $\mathcal{Q}_0^{\mathrm{i}}$ and $\mathcal{Q}_0^{\mathrm{d}}$ by means of the partial meadow constructions introduced in Section 5. Thus, we have five algebras related to rational numbers:

$$\begin{array}{lll} \mathcal{Q}_0^{\mathrm{i}}\,, & \mathcal{Q}_0^{\mathrm{d}}\,, & \\ 0^{-1} \uparrow \mathcal{Q}_0^{\mathrm{i}}\,, & Q \,/\, 0 \uparrow \mathcal{Q}_0^{\mathrm{d}}\,, & (Q \setminus \{0\}) \,/\, 0 \uparrow \mathcal{Q}_0^{\mathrm{d}}\,. \end{array}$$

These algebras have been obtained by means of the well-known initial algebra construction and, in the case of the partial algebras, a simple partial algebra construction. This implies that in all cases only equational logic for total algebras has been used as a logical tool for their construction. The approach followed here contrasts with the usual approach where a special logic for partial algebras would be used for their construction (see e.g. [13]).

We believe that many complications and unclarities in the development of the theories of the partial algebras are avoided by not using some logic of partial functions as a logical tool for their construction. Having constructed $0^{-1} \uparrow \mathcal{Q}_0^{\mathrm{i}}$ in the way described above, the question whether it satisfies the equation $0^{-1} = 0^{-1}$ and related questions are still open because the logic of partial functions to be used when working with $0^{-1} \uparrow \mathcal{Q}_0^{\mathrm{i}}$ has not been fixed yet. This means that it is still a matter of design which logic of partial functions will be used when working with this partial algebra. As soon as the logic is fixed, the above-mentioned questions are no longer open: it is anchored in the logic whether $0^{-1} = 0^{-1}$ is satisfied, $0^{-1} \neq 0^{-1}$ is satisfied, or neither of the two is satisfied. Similar remarks apply to the other two partial algebras obtained above.

Many people prefer $0^{-1} \uparrow \mathcal{Q}_0^{\mathrm{i}}$ to the other algebras related to rational numbers. It is likely that this is because $x \cdot x^{-1} = 1$ serves as an implicit definition of ${}^{-1}$ in $0^{-1} \uparrow \mathcal{Q}_0^{\mathrm{i}}$.

To prove that $\mathcal{Q}_0^{\mathrm{i}}$ differs from the field of rational numbers only in that the multiplicative inverse of zero is zero, we need some auxiliary results.

Lemma 1. *Let p be a prime number. Then for each $u \in \mathbb{Z}_p$, there exists $v, w \in \mathbb{Z}_p$ such that $u = v^2 + w^2$.*

Proof. The case where $p = 2$ is trivial. In the case where $p \neq 2$, p is odd, say $2 \cdot n + 1$. Let S be the set $\{u \in \mathbb{Z}_p \mid \exists v \in \mathbb{Z}_p \bullet u = v^2\}$, and let $c \in \mathbb{Z}_p$ be such that $c \notin S$. Because $0 \in \mathbb{Z}_p$ and each element of S has at most two roots, we have $|S| \geq n + 1$. For each $u \in c \cdot S$, $u = 0$ or $u \notin S$, as $u \neq 0$ and $u \in S$ only if $c \in S$. Because $c \cdot u \neq c \cdot v$ for each $u, v \in S$ with $u \neq v$, we have $|c \cdot S| \geq n + 1$. It follows that $S \cup c \cdot S = \mathbb{Z}_p$ and $S \cap c \cdot S = \{0\}$. This implies that $c \cdot S = \{u \in \mathbb{Z}_p \mid \forall v \in \mathbb{Z}_p \bullet u \neq v^2\} \cup \{0\}$. Hence, for each $u \in \mathbb{Z}_p$ with $u \notin S$, there exists an $v \in \mathbb{Z}_p$ such that $u = c \cdot v^2$. The set S is not closed under sums, as $1 \in S$, and every element of $\mathbb{Z}_p$ is a sum of ones. This implies that there exist $u, v \in \mathbb{Z}_p$ such that $u^2 + v^2 \notin S$. Let $a, b \in \mathbb{Z}_p$ be such that $a^2 + b^2 \notin S$, and take $a^2 + b^2$ for c. Then for each $u \in \mathbb{Z}_p$ with $u \notin S$, there exists an $v \in \mathbb{Z}$ such that $u = (a^2 + b^2) \cdot v^2$. Because $(a^2 + b^2) \cdot v^2 = (a \cdot v)^2 + (b \cdot v)^2$, we have that, for each $u \in \mathbb{Z}_p$ with $u \notin S$, there exist $v, w \in \mathbb{Z}$ such that $u = v^2 + w^2$. Because $u \in S$ iff $u = v^2 + 0^2$ for some $v \in \mathbb{Z}_p$, we have that, for each $u \in \mathbb{Z}_p$ with $u \in S$, there exist $v, w \in \mathbb{Z}$ such that $u = v^2 + w^2$. $\square$

Corollary 1. *Let p be a prime number. Then there exists $u, v, w \in \mathbb{N}$ such that $w \cdot p = u^2 + v^2 + 1$.*

Proof. By Lemma 1, there exist $u, v \in \mathbb{Z}_p$ such that $-1 = u^2 + v^2$. Let $a, b \in \mathbb{Z}_p$ be such that $-1 = a^2 + b^2$. Then $a^2 + b^2 + 1$ is a multiple of p in $\mathbb{N}$. Hence, there exists $u, v, w \in \mathbb{N}$ such that $w \cdot p = u^2 + v^2 + 1$. $\square$

Theorem 3. *$\mathcal{Q}_0^{\mathrm{i}} = I(\Sigma_{\mathrm{imd}}, E_{\mathrm{imd}} \cup \{(1 + x^2 + y^2) \cdot (1 + x^2 + y^2)^{-1} = 1\})$ is the zero-totalized field of rational numbers, i.e. the Σ_{imd}-algebra that differs from the field of rational numbers only in that $0^{-1} = 0$.*

Proof. From the proof of Theorem 3.6 from [10], we already know that, for each set E' of Σ_{imd}-equations valid in the zero-totalized field of rational numbers, $I(\Sigma_{\mathrm{imd}}, E_{\mathrm{imd}} \cup E')$ is the zero-totalized field of rational numbers if it follows from $E_{\mathrm{imd}} \cup E'$ that $\underline{u}$ has a multiplicative inverse for each $u \in \mathbb{N}$. Because $1 + x^2 + y^2 \neq 0$, we have that $(1 + x^2 + y^2) \cdot (1 + x^2 + y^2)^{-1} = 1$ is valid in the zero-totalized field of rational numbers. So it remains to be proved that $\underline{u}$ has a multiplicative inverse for each $u \in \mathbb{N}$.

Let p be a prime number. Then, by Corollary 1, there exist $u, v, w \in \mathbb{N}$ such that $w \cdot p = u^2 + v^2 + 1$. Let $m, a, b \in \mathbb{N}$ be such that $m \cdot p = a^2 + b^2 + 1$. It is easily proved by induction that $\underline{u + v} = \underline{u} + \underline{v}$ and $\underline{u \cdot v} = \underline{u} \cdot \underline{v}$ for all $u, v \in \mathbb{N}$ in any meadow. It follows that $\underline{m} \cdot \underline{p} = \underline{a}^2 + \underline{b}^2 + \underline{1}$ in $\mathcal{Q}_0^{\mathrm{i}}$. Because $(1 + x^2 + y^2) \cdot (1 + x^2 + y^2)^{-1} = 1$ in $\mathcal{Q}_0^{\mathrm{i}}$, we have $(\underline{m} \cdot \underline{p}) \cdot (\underline{m} \cdot \underline{p})^{-1} = 1$. This implies that $\underline{m} \cdot (\underline{m} \cdot \underline{p})^{-1}$ is the multiplicative inverse of $\underline{p}$. Hence, $\underline{u}$ has a multiplicative inverse for each $u \in \mathbb{N}$ that is a prime number. Let $c \in \mathbb{N}$. Then c is the product of finitely many prime numbers, say $p_1 \cdot \cdots \cdot p_n$. Because $(\underline{p_1} \cdot \cdots \cdot \underline{p_n})^{-1} = \underline{p_1}^{-1} \cdot \cdots \cdot \underline{p_n}^{-1}$ in any meadow (see e.g. Proposition 2.8 in [5]) and $\underline{c} = \underline{p_1} \cdot \cdots \cdot \underline{p_n}$, we have that $\underline{p_1}^{-1} \cdot \cdots \cdot \underline{p_n}^{-1}$ is the multiplicative inverse of $\underline{c}$. Hence, $\underline{u}$ has a multiplicative inverse for each $u \in \mathbb{N}$. $\square$

Lemma 1, Corollary 1, and Theorem 3 come from Hirshfeld (personal communication, 31 January 2009). Lemma 1 is a folk theorem in the area of field theory, but we could not find a proof of it in the literature.

We remark that in [10], the initial algebra specification of $\mathcal{Q}_0^{\mathrm{i}}$ is obtained by adding the equation $(1+x^2+y^2+z^2+w^2) \cdot (1+x^2+y^2+z^2+w^2)^{-1} = 1$ instead of the equation $(1+x^2+y^2) \cdot (1+x^2+y^2)^{-1} = 1$ to E_{imd}. In other words, in the current paper, we have reduced the number of squares needed in the equation added to E_{imd} from 4 to 2. In [6], it is shown that the number of squares cannot be reduced to 1.

7 Two-Valued Logics of Partial Functions

In this section, we take a survey of two-valued first-order logics that may be used when working with partial algebras such as $0^{-1} \uparrow \mathcal{Q}_0^{\mathrm{i}}$, $Q \mathbin{/} 0 \uparrow \mathcal{Q}_0^{\mathrm{d}}$, and $(Q \setminus \{0\}) \mathbin{/} 0 \uparrow \mathcal{Q}_0^{\mathrm{d}}$, focussing on semantics.

We consider two truth values, which correspond to true and false and which are denoted by T and F, respectively.

In handling partial functions in a two-valued logic, it is possible not to deviate essentially from classical first-order logic because partial functions can be represented by total functions if the domain of the structure in question is extended with an undefined value. In this way, non-denoting terms do not occur: the terms in question denote the undefined value. A serious drawback of this approach is that the undefined value is not treated differently from the ordinary values. For instance, quantification is over ordinary values as well as the undefined value. As a result, frequent reasoning about undefined is customary. This approach is followed in e.g. Scott's logic of computable functions [16]. Henceforth, we will use the term two-valued logics of partial functions exclusively for logics that deviate essentially from classical first-order logic.

Two-valued logics of partial functions do not give up the excluded middle: when a formula cannot be classified as true, it is inexorably classified as false. In general, atomic formulas in which non-denoting terms occur are classified as false. Below, we will mention one exception: equations in the case where strong equality is taken as basic.

The key differences between the various two-valued logics of partial functions are with respect to:

– how free variables and bound variables are treated;
– what kind of equality is taken as basic;
– what additional predicates are taken as basic.

There are two ways of treating free variables and bound variables in two-valued logics of partial functions:

– free variables may not denote, but bound variables always do;
– both bound variables and free variables always denote.

The first treatment has the advantage that frequent reasoning about the definedness of terms can be avoided. It is found in e.g. Scott's free logic [31] and MPL_ω [24]. The second treatment keeps a logic closer to classical first-order logic than the first treatment. It is found in e.g. LPT [2].

There are two ways of adapting equality to non-denoting terms in two-valued logics. The two ways in question lead to kinds of equality which only differ in how non-denoting terms are handled:

– *strong equality*: if either t or t' is non-denoting, then the truth value of $t = t'$ is T whenever both t and t' are non-denoting and F otherwise;
– *existential equality*: if either t or t' is non-denoting, then the truth value of $t = t'$ is F.

Strong equality can be expressed in terms of existential equality (see e.g. [13]).

The definedness predicate is virtually the only additional predicate that is taken as basic in two-valued logics of partial functions. This unary predicate is defined as follows: $D(t) = \mathsf{T}$ if t is denoting, and $D(t) = \mathsf{F}$ otherwise. Definedness can be expressed in terms of existential equality. Moreover, existential equality can be expressed in terms of definedness and strong equality (see e.g. [13]).

In most two-valued logics of partial functions that have been proposed, including Scott's free logic, MPL_ω and LPT, existential equality and definedness are taken as basic.

8 Three-Valued Logics of Partial Functions

In this section, we take a survey of three-valued first-order logics that may be used when working with partial algebras such as $0^{-1} \uparrow \mathcal{Q}_0^{\mathrm{i}}$, $Q \mathbin{/} 0 \uparrow \mathcal{Q}_0^{\mathrm{d}}$, and $(Q \setminus \{0\}) \mathbin{/} 0 \uparrow \mathcal{Q}_0^{\mathrm{d}}$, focussing on semantics.

We consider three truth values, which correspond to true, false, and neither-true-nor-false and which are denoted by T, F, and $*$, respectively.

The key differences between the various three-valued logics of partial functions are with respect to:

- how the classical logical connectives and quantifiers are extended to the three-valued case;
- what additional logical connectives are taken as basic;
- what kind of equality is taken as basic;
- what model-theoretic notion of logical consequence is taken to underlie the proof system.

In three-valued logics of partial functions, the classical logical connectives are always extended to total three-valued truth functions (as opposed to partial three-valued truth functions) and the classical logical quantifiers correspondingly.

Even if we require the usual interdefinability of the classical logical connectives, there remain 3^6 ways of extending them to the three-valued case. However, many ways must be considered uninteresting for a logic of partial functions because they lack an interpretation of the third truth value that fits in with its origin: dealing with non-denoting terms. If those ways are excluded, only four ways to extend the classical logical connectives to the three-valued case remain (see e.g. [3]). Three of them are well-known: they lead to Bochvar's strict connectives [12], McCarthy's sequential connectives [25], and Kleene's monotonic connectives [22]. The fourth way leads to McCarthy's sequential connectives with the role of the operands of the binary connectives reversed. For each of these kinds of connectives, the quantifiers belonging to it are the natural generalizations of the conjunction and disjunction in question. The quantifiers belonging to McCarthy's sequential connectives, which are relatively unknown, are considered in [23].

In all above-mentioned ways of extending the classical logical connectives, the resulting connectives are monotonic with respect the ordering $\preceq$ on three-valued domain of truth values defined as follows: $b \preceq b'$ iff $b \neq *$ implies $b = b'$. Kleene's connectives are the strongest monotonic extensions of the classical connectives. They are expressively complete for all monotonic three-valued truth functions. This implies that Bochvar's connectives and McCarthy's connectives can be expressed in terms of Kleene's connectives. Hence, only the addition of McCarthy's conjunction and disjunction to Bochvar's connectives and the addition of non-monotonic connectives to Bochvar's connectives, McCarthy's connectives or Kleene's connectives make sense. The most interesting non-monotonic connective appears to be Hoogewijs' definedness connective [19], which is defined as follows: $\Delta b = \mathsf{T}$ if not $b = *$, and $\Delta b = \mathsf{F}$ otherwise. Kleene's connectives together with this definedness connective are expressively complete for all three-valued truth functions.

There are only three ways of extending classical equality to the three-valued case that possess an interpretation of the third truth value that fits in with dealing with non-denoting terms. The three ways in question lead to kinds of equality which only differ in their treatment of non-denoting terms:

– *weak equality*: if either t or t' is non-denoting, then the truth value of $t = t'$ is $*$;
– *strong equality*: if either t or t' is non-denoting, then the truth value of $t = t'$ is T whenever both t and t' are non-denoting and F otherwise;
– *existential equality*: if either t or t' is non-denoting, then the truth value of $t = t'$ is F.

Kleene's connectives, the quantifiers belonging to them, and weak equality match very well together: each behaves according to its classical truth-condition and falsehood-condition; only if neither of them meets, it will yield $*$. Weak equality is taken as basic in many three-valued logics of partial functions in which also Kleene's connectives and the quantifiers belonging to them are taken as basic. Strong equality and existential equality correspond with the two kinds of equality that are considered in two-valued logics of partial functions.

In the presence of Kleene's connectives and the definedness connective mentioned above, strong equality and existential equality can be expressed in terms of weak equality (see e.g. [27]).

The following are the intuitive ideas that underlie the most sensible notions of logical consequence for three-valued logics:

– from premises that are true, one can draw conclusions that are true;
– from premises that are true, one can draw conclusions that are not false;
– from premises that are not false, one can draw conclusions that are true;
– from premises that are not false, one can draw conclusions that are not false.

The notions of logical consequence that correspond with these ideas are called *ss-consequence*, *sw-consequence*, *ws-consequence*, and *ww-consequence*, respectively (s stands for strong and w stands for weak). For formulas formed with Kleene's connectives and the quantifiers belonging to them (cf. [23]):

– ss-consequence amounts to classical logical consequence except for the absence of what depends anyhow on the excluded middle;
– sw-consequence amounts to classical logical consequence;
– ws-consequence amounts to ss-consequence with the difference that premises that are neither true nor false are ignored;
– ww-consequence amounts to sw-consequence with the difference that premises that are neither true nor false are ignored.

We see that ss-consequence is lacking exactly what does not leave room for formulas which are neither true nor false. This is what one naturally expects from a three-valued logic where the additional truth value is interpreted as neither true nor false. Each of the above-mentioned notions of logical consequence underlies known three-valued logics of partial functions, e.g. ss-consequence underlies LPF [1], sw-consequence underlies PPC [19] and PFOL [15], ws-consequence underlies WS [29], and ww-consequence underlies WL [28].

For formulas formed with Kleene's connectives, the quantifiers belonging to them, and the definedness connective mentioned above, each of these notions of logical consequence can be used to define the others (see e.g. [15]).

In most three-valued logics of partial functions that have been proposed, including LPF, PPC, PFOL and WS, Kleene connectives, the quantifiers belonging to them, Hoogewijs' definedness connective and weak equality are taken as basic.

9 Discussion of Logics of Partial Functions

In Sections 7 and 8, it is not only mentioned what the key differences are between the various two-valued logics of partial functions and what the key differences are between the various three-valued logics of partial functions. It is also mentioned what their main connections are. From that, it is clear that are a few logics which together encompass all others. However, this should not be taken as an indication that only those few logics should be considered when fixing a logic to be used for a particular purpose, such as reasoning about some partial meadow of rational numbers. Whether a logic of partial functions is suited to a particular purpose may depend on quite different issues.

It follows from results in [21, 30] how the few logics of partial functions which together encompasses all others can be reconstructed classically by embeddings into classical logic. Of course, this should not be taken as an indication that we should stick to classical logic either.

Using Kleene's connectives, the quantifiers belonging to them, and weak equality, formulas can easily be understood. A formula is classified as true or false according to the classical truth-condition and falsehood-condition for the logical connectives, quantifiers, and equality; it is classified as neither-true-nor-false exactly when it cannot be classified as true or false by these conditions. The notion of logical consequence that fits in most naturally with this is ss-consequence, because it is the only one that is lacking exactly what does not leave room for formulas which are neither true nor false. However, it is not easy to memorize the logical axioms and inference rules of a proof system for ss-consequence if we are used to classical logic.

Formulas that can easily be understood and an easily memorizable proof system are not the only matters for consideration when fixing a logic. Another important matter is that the validity of formulas in the partial meadow of rational numbers under consideration is in accordance with the intuition. For instance, in the case of a three-valued logic of partial functions with Bochvar's, McCarthy's or Kleene's connectives, the quantifiers belonging to the connectives in question, and weak equality, the formula $\forall x \bullet (x \,/\, x = 1 \Rightarrow x \neq 0)$ is not true in $Q \,/\, 0 \uparrow \mathcal{Q}_0^{\mathrm{d}}$, which is likely to be considered counterintuitive.

Sections 7 and 8 show that there are many matters about which a choice has to be made when designing a logic of partial functions to be used when working with a partial meadow of rational numbers. The logics of partial functions that have been proposed in the past cover only a small part of the possible combinations of choices. It is not clear to us what is at the root of this. Therefore, we do no preclude the possibility that other logics of partial functions are also worth consideration.

10 Conclusions

We have made a formal distinction between inverse meadows and divisive meadows, have given a translation from the terms over the signature of divisive meadows into the terms over the signature of inversive meadows and a translation the other way round, and have shown, using those translations, that it depends on the angle from which they are viewed whether inversive meadows or divisive meadows must be considered more basic. We have also introduced simple constructions of variants of inverse meadows and divisive meadows with a partial multiplicative inverse or division operation.

We have obtained five algebras related to rational numbers by means of the well-known initial algebra construction and, in three cases, the above-mentioned partial algebra constructions. This implies that in all cases only equational logic for total algebras has been used as a logical tool for their construction. In this way, we have avoided choosing or developing an appropriate logic, which we consider a design problem of logics, not of data types. We have also taken a survey of first-order logics that are appropriate to handle those partial variants of inversive and divisive meadows.

The survey of logics of partial functions that we have taken is a survey of options. It does not include a comparison of their usefulness for a particular purpose. Such comparisons are found in e.g. [14, 20]. The problem with them is that they are based on selected examples, and therefore they cover only a few forms of formulas and a few patterns of proofs. An option for future work is to carry out a more systematic comparison of the different logics of partial functions by which the partial variants of inversive and divisive meadows can be handled, with the aim to develop a clear picture of their differences.

We claim that, viewed from the theory of abstract data types, the way in which partial algebras are constructed in this paper is the preferred way. Its main advantage is that no decision need to be taken in the course of the construction about matters concerning the logic to be used when working with the partial algebras in question. For that reason, we consider it useful to generalize the partial algebra constructions on inversive and divisive meadows to a partial algebra construction that can be applied to any total algebra. This is another option for future work.

The axioms of an inversive meadow forces that the equation $0^{-1} = 0$ holds. It happens that this equation is used for technical convenience in several other places, see e.g. [18, 17]. The axioms of a divisive meadow forces that the equation $x \,/\, 0 = 0$ holds. One of the few published pieces of writing about this equation that we have been able to trace is [26].

References

1. Barringer, H., Cheng, J.H., Jones, C.B.: A logic covering undefinedness in program proofs. Acta Informatica **21**(3), 251–269 (1984)
2. Beeson, M.J.: Proving programs and programming proofs. In: R. Barcan-Marcus, G.J.W. Dorn, P. Weingartner (eds.) Logic, Methodology, and Philosophy of Science VII, pp. 51–81. North-Holland, Amsterdam (1986)
3. Bergstra, J.A., Bethke, I., Rodenburg, P.H.: A propositional logic with 4 values: True, false, divergent and meaningless. Journal of Applied Non-Classical Logic **5**(2), 199–218 (1995)
4. Bergstra, J.A., Heering, J., Klint, P.: Module algebra. Journal of the ACM **37**(2), 335–372 (1990)
5. Bergstra, J.A., Hirshfeld, Y., Tucker, J.V.: Meadows and the equational specification of division. Theoretical Computer Science **410**(12–13), 1261–1271 (2009)
6. Bergstra, J.A., Hirshfeld, Y., Tucker, J.V.: Skew meadows. `arXiv:0901.0803 [math.RA]` at `http://arxiv.org/` (2009)
7. Bergstra, J.A., Loots, M.E.: Program algebra for sequential code. Journal of Logic and Algebraic Programming **51**(2), 125–156 (2002)
8. Bergstra, J.A., Ponse, A.: A generic basis theorem for cancellation meadows. `arXiv:0803.3969 [math.RA]` at `http://arxiv.org/` (2008)
9. Bergstra, J.A., Tucker, J.V.: Algebraic specifications of computable and semicomputable data types. Theoretical Computer Science **50**(2), 137–181 (1987)
10. Bergstra, J.A., Tucker, J.V.: The rational numbers as an abstract data type. Journal of the ACM **54**(2), Article 7 (2007)
11. Bethke, I., Rodenburg, P.H., Sevenster, A.: The structure of finite meadows. `arXiv:0903.1196 [cs.LO]` at `http://arxiv.org/` (2009)
12. Bochvar, D.A.: On a three-valued logical calculus and its application to the analysis of contradictions (in Russian). Matématičeskij Sbornik **4**(46), 287–308 (1939). A translation into English appeared in History and Philosophy of Logic **2**(1–2), 87–112 (1981)
13. Cerioli, M., Mossakowski, T., Reichel, H.: From total equational to partial first order logic. In: E. Astesiano, H.J. Kreowski, B. Krieg-Brückner (eds.) Algebraic Foundations of Systems Specification, pp. 31–104. Springer-Verlag, Berlin (1999)

14. Cheng, J.H., Jones, C.B.: On the usability of logics which handle partial functions. In: C. Morgan, J.C.P. Woodcock (eds.) 3rd Refinement Workshop, Workshops in Computing Series, pp. 51–69. Springer-Verlag (1991)
15. Gavilanes-Franco, A., Lucio-Carrasco, F.: A first order logic for partial functions. Theoretical Computer Science **74**(1), 37–69 (1990)
16. Gordon, M.J.C., Milner, R., Wadsworth, C.: Edinburgh LCF, *Lecture Notes in Computer Science*, vol. 78. Springer-Verlag, Berlin (1979)
17. Harrison, J.: Theorem Proving with the Real Numbers. Distinguished Dissertations Series. Springer-Verlag, Berlin (1998)
18. Hodges, W.A.: Model Theory, *Encyclopedia of Mathematics and Its Applications*, vol. 42. Cambridge University Press, Cambridge (1993)
19. Hoogewijs, A.: A calculus of partially defined predicates. Mathematical scripts, Rijksuniversiteit Gent (1977)
20. Jones, C.B.: Reasoning about partial functions in the formal development of programs. Electronic Notes in Theoretical Computer Science **145**, 3–25 (2006)
21. Jones, C.B., Middelburg, C.A.: A typed logic of partial functions reconstructed classically. Acta Informatica **31**(5), 399–430 (1994)
22. Kleene, S.C.: On notation for ordinal numbers. Journal of Symbolic Logic **3**(4), 150–155 (1938)
23. Konikowska, B., Tarlecki, A., Blikle, A.: A three-valued logic for software specification and validation. Fundamenta Informaticae **14**, 411–453 (1991)
24. Koymans, C.P.J., Renardel de Lavalette, G.R.: The logic MPL$_\omega$. In: M. Wirsing, J.A. Bergstra (eds.) Algebraic Methods: Theory, Tools and Applications, *Lecture Notes in Computer Science*, vol. 394, pp. 247–282. Springer-Verlag (1989)
25. McCarthy, J.: A basis for a mathematical theory of computation. In: P. Braffort, D. Hirschberg (eds.) Computer Programming and Formal Systems, pp. 33–70. North-Holland, Amsterdam (1963)
26. McDonnell, E.E.: Zero divided by zero. In: APL '76, pp. 295–296. ACM Press (1976)
27. Middelburg, C.A., Renardel de Lavalette, G.R.: LPF and MPL$_\omega$ – a logical comparison of VDM SL and COLD-K. In: S. Prehn, W.J. Toetenel (eds.) VDM '91, Volume 1, *Lecture Notes in Computer Science*, vol. 551, pp. 279–308. Springer-Verlag, Berlin (1991)
28. Owe, O.: An approach to program reasoning based on a first order logic for partial functions. Technical Report 89, Institute of Informatics, University of Oslo (1985)
29. Owe, O.: Partial logics reconsidered: A conservative approach. Formal Aspects of Computing **5**(3), 208–223 (1993)
30. Renardel de Lavalette, G.R.: The static part of the design language COLD-K. In: D.J. Andrews, J.F. Groote, C.A. Middelburg (eds.) Semantics of Specification Languages, Workshops in Computing Series, pp. 51–82. Springer-Verlag (1994)
31. Scott, D.S.: Existence and description in formal logic. In: R. Schoenman (ed.) Bertrand Russell, Philosopher of the Century, pp. 181–200. Allen & Unwin (1967)
32. Verloren van Themaat, W.A.: Right-divisive groups. Notre Dame Journal of Formal Logic **19**(1), 137–140 (1978)
33. Visser, A.: Categories of theories and interpretations. In: A. Enayat, I. Kalantari, M. Moniri (eds.) Logic in Tehran 2003, *Lecture Notes in Logic*, vol. 26, pp. 284–341. Association for Symbolic Logic (2006)
34. Yamada, M.: Inversive semigroups I. Proceedings of Japan Academy **39**(2), 100–103 (1963)

Electronic Reports Series of the Programming Research Group

[PRG0713] J.A. Bergstra, A. Ponse, and M.B. van der Zwaag, *Tuplix Calculus*, Programming Research Group - University of Amsterdam, 2007.

[PRG0712] J.A. Bergstra, S. Nolst Trenite, and M.B. van der Zwaag, *Towards a Formalization of Budgets*, Programming Research Group - University of Amsterdam, 2007.

[PRG0711] J.A. Bergstra and C.A. Middelburg, *Program Algebra with a Jump-Shift Instruction,* Programming Research Group - University of Amsterdam, 2007.

[PRG0710] J.A. Bergstra and C.A. Middelburg, *Instruction Sequences with Dynamically Instantiated Instructions*, Programming Research Group - University of Amsterdam, 2007.

[PRG0709] J.A. Bergstra and C.A. Middelburg, *Instruction Sequences with Indirect Jumps,* Programming Research Group - University of Amsterdam, 2007.

[PRG0708] B. Diertens, *Software (Re-)Engineering with PSF III: an IDE for PSF*, Programming Research Group - University of Amsterdam, 2007.

[PRG0707] J.A. Bergstra and C.A. Middelburg, *An Interface Group for Process Components,* Programming Research Group - University of Amsterdam, 2007.

[PRG0706] J.A. Bergstra, Y. Hirschfeld, and J.V. Tucker, *Skew Meadows,* Programming Research Group - University of Amsterdam, 2007.

[PRG0705] J.A. Bergstra, Y. Hirschfeld, and J.V. Tucker, *Meadows*, Programming Research Group - University of Amsterdam, 2007.

[PRG0704] J.A. Bergstra and C.A. Middelburg, *Machine Structure Oriented Control Code Logic (Extended Version),* Programming Research Group - University of Amsterdam, 2007.

[PRG0703] J.A. Bergstra and C.A. Middelburg, *On the Operating Unit Size of Load/Store Architectures*, Programming Research Group - University of Amsterdam, 2007.

[PRG0702] J.A. Bergstra and A. Ponse, *Interface Groups and Financial Transfer Architectures,* Programming Research Group - University of Amsterdam, 2007.

[PRG0701] J.A. Bergstra, I. Bethke, and M. Burgess, *A Process Algebra Based Framework for Promise Theory*, Programming Research Group - University of Amsterdam, 2007.

[PRG0610] J.A. Bergstra and C.A. Middelburg, *Parallel Processes with Implicit Computational Capital*, Programming Research Group - University of Amsterdam, 2006.

[PRG0609] B. Diertens, *Software (Re-)Engineering with PSF II: from architecture to implementation*, Programming Research Group - University of Amsterdam, 2006.

[PRG0608] A. Ponse and M.B. van der Zwaag, *Risk Assessment for One-Counter Threads,* Programming Research Group - University of Amsterdam, 2006.

[PRG0607] J.A. Bergstra and C.A. Middelburg, *Synchronous Cooperation for Explicit Multi-Threading*, Programming Research Group - University of Amsterdam, 2006.

[PRG0606] J.A. Bergstra and M. Burgess, *Local and Global Trust Based on the Concept of Promises*, Programming Research Group - University of Amsterdam, 2006.

[PRG0605] J.A. Bergstra and J.V. Tucker, *Division Safe Calculation in Totalised Fields*, Programming Research Group - University of Amsterdam, 2006.

[PRG0604] J.A. Bergstra and A. Ponse, *Projection Semantics for Rigid Loops*, Programming Research Group - University of Amsterdam, 2006.

[PRG0603] J.A. Bergstra and I. Bethke, *Predictable and Reliable Program Code: Virtual Machine-based Projection Semantics (submitted for inclusion in the Handbook of Network and Systems Administration),* Programming Research Group - University of Amsterdam, 2006.

[PRG0602] J.A. Bergstra and A. Ponse, *Program Algebra with Repeat Instruction*, Programming Research Group - University of Amsterdam, 2006.