Abstract Data Types: Stacks, Fields, and Meadows

Jan Bergstra

Section Theoretical Computer Science
Informatics Institute, Faculty of Science
University of Amsterdam
j.a.bergstra@uva.nl

Oktober 16, 2013
My research topics in order of (subjective) “priority”:

1. **Meadows** [because (i) these structures are “everywhere”, and (ii) the story is so clear].

2. **Promises** [following the approach of Mark Burgess (Oslo), because promises are so common and so little work has been done about them].
My research topics in order of (subjective) “priority”:

1. **Meadows** [because (i) these structures are “everywhere”, and (ii) the story is so clear].

2. **Promises** [following the approach of Mark Burgess (Oslo), because promises are so common and so little work has been done about them].

3. **Informational money** [because Bitcoin is “the way ahead for money”].

4. **Instruction sequences** [because computer programs are the essential modern industrial product type, and I therefore want to develop a theory of programs “from scratch”].
My research topics in order of (subjective) “priority”:

1. **Meadows** [because (i) these structures are “everywhere”, and (ii) the story is so clear].

2. **Promises** [following the approach of Mark Burgess (Oslo), because promises are so common and so little work has been done about them].

3. **Informational money** [because Bitcoin is “the way ahead for money”].

4. **Instruction sequences** [because computer programs are the essential modern industrial product type, and I therefore want to develop a theory of programs “from scratch”].

5. **Decision taking** [because I don’t “believe” (like, accept, appreciate) current theories of decision taking, and yet DT is of critical importance].

6. **Short-circuit logic** [because it appears in every program notation and there is no existing theory of it yet (except our own work)].
Abstract data types: dogma’s from the early 80’s

Algebraic specifications of abstract data types; focus on initial algebras

Signatures: sets of symbols (names) for constants, functions, and sorts

- non-empty sorts (first order logic)
- total functions (simplified type-system), no subsorts,
- focus on minimal algebras (each object is the meaning of an expression).

EXAMPLE. Stack of $A$ (sort name denoting a set of objects often written as $\| A \|$) and a sort named $S$ (collection of stacks over $\| A \|$, often written $\| A \|^*$)

- Constants: $\emptyset$ of sort $S$ (empty stack),
- Functions: $\text{push} : A \times S \rightarrow S$, $\text{pop} : S \rightarrow S$, and $\text{top} : S \rightarrow A$. 

Jan Bergstra (STCS)  
meadows and division by zero  
Oktober 16, 2013 3 / 30
Equations for stacks (and specification of the stack data type):

**two obvious cases**
\[
\text{top}(\text{push}(a, s)) = a \\
\text{pop}(\text{push}(a, s)) = s
\]

**one less obvious (but workable) case**
\[
\text{pop}(\emptyset) = \emptyset
\]

**one difficult case**  \(\text{top}(\emptyset) = a\) (with \(a\) a new constant of sort \(A\))
(with \(a\) the name of some special \(a\) in \(\|A\|\). Here \(a\) serves the role of an error element; \(a\) names the error)

an unsatisfactory consequence? (Stack data type not fully abstract.)

Consider \(p = \emptyset\) and \(q = \text{push}(a, \emptyset)\) then:
\[
\text{top}(p) = a = \text{top}(q), \text{ and} \\
\text{pop}(p) = \emptyset = \text{pop}(q)
\]

Interesting option (fully abstract stack)
\[
\emptyset = \text{push}(a, \emptyset) \text{ (that is: } p = q)\).
Thus: \(\emptyset = \text{push}(a, \emptyset) = \text{push}(a, \text{push}(a, \emptyset)) = \ldots\)
Other abstract data types

- **numbers**: boolean, natural, integer, float.
- **sets**: finite sets, multi-sets.
- **tables**: hashtables, arrays.
- **trees**: AVL-trees, Quad-trees, BDDs.
- **grammars**: entire grammars of language syntax (ASF+SDF, Rascal).
- **crypto**: String sets with encryption an decryption functions.
- **OO**: classes with static methods only.
Are you familiar (need not be very familiar) with:

- integers ($\mathbb{N}$) (yes/no)
- rational numbers ($\mathbb{Q}$), (yes/no)
- real numbers ($\mathbb{R}$), (yes/no)
- complex numbers ($\mathbb{C}$) (yes/no).

Further questions are about rational numbers (that is $\mathbb{Q}$):

2. Simplification of $\frac{X \cdot X \cdot X}{X}$ results in $X \cdot X$ (yes/no).

3. $\frac{1}{0} = \frac{1+1}{0}$ (yes/no).

4. $\frac{(1+1) \cdot (1-1)}{((1+1)-1)-1}$ does not exist (yes/no).

5. $\frac{(1+1) \cdot (1-1)}{((1+1)-1)-1} = \frac{(1+1) \cdot 0}{1-1}$ (yes/no).

6. The question “what is $\frac{1}{0}$?” is a meaningless question (yes/no).
1. Bayes’ rule is as follows: \( P(x \mid e) = \frac{P(e \mid x) \cdot P(x)}{P(e)} \) provided \( P(e) \neq 0 \) (yes/no).

2. Division is a partial function (yes/no).

3. If division is a partial function then:
   - The question “what is \( \frac{1}{0} \)” is a meaningful question (yes/no), and
   - \( \frac{1}{0} \) is undefined (yes/no), and
   - \( \frac{1}{0} \neq \frac{1+1}{0} \) (yes/no).

4. \( \div \) is a function symbol (yes/no).

5. If \( \div \) is a function symbol then division (its meaning) is a partial function (yes/no).

6. Is \( \frac{p}{q} \) an abbreviation for “the unique \( r \) such that \( r \cdot q = p \)”? (yes/no).

7. Is \( \frac{1+1}{1} \) an abbreviation for “the unique \( r \) such that \( r \cdot 1 = 1 + 1 \)” (yes/no).

8. Is \( \frac{1+1}{0} \) an abbreviation (yes/no).

9. Is \( \frac{p}{q} \) an abbreviation for “the unique \( r \) such that \( r \cdot q = p \)” provided that \( q \neq 0 \) (yes/no).
1. In reply to the question: “simplify $\frac{X \cdot X \cdot Y}{X \cdot X}$”, a correct answer is $\frac{X \cdot Y}{X}$?

2. Consider the question Q:

$$Q \equiv \text{“ is the following equality valid } \frac{(1 + 1) + (1 + 1)}{(1 + 1) - (1 + 1)} = \frac{((1 + 1) + 1) + 1}{((1 + 1) - 1) - 1}?\text{”}.$$ 

About Q there are these questions:

1. Is it permissible for an NL math teacher to ask Q (level say 4 VO) (yes/no)?

2. (If “yes”):

   1. Is it permissible for a math teacher to consider the reply “yes” to Q valid?

   2. Is it permissible for a math teacher to consider the reply “yes” to Q invalid?

3. (If “no”, that is Q is not a permissible question:)

   1. How can a teacher express the knowledge that Q should not be asked?

   2. Is the student supposed to have that knowledge as well?
PROPOSAL: organization of elementary mathematics along the lines of abstract data types.

- view Rationals, Reals, and Complex Numbers as ADTs.
- Explicit definition of syntax versus meaning (semantics):
  - syntax: terms, equations, provability, proofs
  - semantics: model, interpretation, value function, domain, truth.
- Clear viewpoints about signatures: which sorts, constants and functions.
- Clear viewpoints about named abbreviations (derived operators).
- Clear viewpoints about naming of expressions.

“NEW”: division viewed as a function symbol. (Precisely: inverse $[x^{-1}]$ a function symbol, both division notations $[x/y]$, $[\frac{x}{y}]$ as derived function symbols).
FIELD SIGNATURE: signature for number systems enabling division (fields):

- sort name: $S$, (flexible use of name, other names e.g.: $M, K, L$).
- constants: 0 and 1,
- functions: $+$ (addition, 2-place), $\cdot $ (multiplication, 2-place)
- $−$ (minus, that is additive inverse, 1-place),
- abbreviation (derived function symbol): $x − y = x + (−y)$

ON TOP of field signature:

- $\cdot ^{-1}$ (multiplicative inverse, unary),
- abbreviation (derived function symbol) $\frac{x}{y} = x \cdot y^{-1}$ (division, 2-place).

NAMING: once multiplicative inverse or division has been given the status of a function symbol, we speak of a meadow signature (rather than a ring signature or a field signature).
1. Notion of syntax emerges: expression, equation, form.
2. Concept of proof primarily related to syntax.
3. Meaning (semantics): structure that provides interpretation to sort names, constant symbols/names and function symbols/names.
4. Connection between syntax and semantics:

\[ K \models p \text{ and } K, \sigma \models p \ (\sigma \text{ assigns values in } || K || \text{ to variables that may occur in } p). \]
Meadows are unfamiliar to mathematicians: What is wrong with $\frac{1}{0}$?

Mathematicians do not accept that syntax precedes semantics. They will not feel any obligation to explain $\frac{1}{0}$ just because it has been written down by someone. The question “what is $\frac{1}{0}$” is considered methodologically flawed for that reason.

By insisting on working with meadows rather than with fields:

1. $\frac{1}{0}$ is a legal expression irrespective of its meaning.
2. the question “what is $\frac{1}{0}$” becomes meaningful, if not unavoidable.
3. the reply “$\frac{1}{0}$ is undefined” is still an option (partial function interpretation of inverse/division).
4. if “$\frac{1}{0}$ is undefined” then $\frac{1}{0} = \frac{1}{0}$ still has three possible answers: yes, no, and logically undefined, a third truth value. (If no then having $\frac{0}{0} = \frac{0}{0}$ is still an option but in any case $X = X$ is not closed under substitution. If yes then $0 \cdot \frac{0}{0} = 0$ is an option that may or may not be ruled out.)
Working with meadows instead of fields. So what is the deal?

Once working with meadows $0^{-1}$ is an expression just as much as say $(1 + 0) \cdot (-1)$. And therefore the following viewpoints/attitudes are ruled out:

- Portraying the question “what is $0^{-1}$” as ridiculous, or stupid, or ill-informed.
- Holding that $0^{-1}$ must (may) not be written (in professional documents).
- Suggesting that the question “what is $0^{-1}$” fails to admit a stable and written answer.
- Claiming that the value of $0^{-1}$ must be undefined.
Importance of signature $\Sigma$ of structure say $F$:

- Signature $\Sigma$ determines the language that can be used to specify $F$ and to reason about it.
- $\Sigma$ determines (degrees of freedom for the design of) proof rules and proof systems for reasoning about $F$. (Essential for verification and validation of programs based on $F$ as an ADT.)
- $\Sigma$ contains the constants and functions that can be used as primitives in program notations operating on basis of (an implementation of) $F$.
- $\Sigma$ determines the collection of expressions that plays a role in term rewriting systems which are used for automated implementation of equational specifications of $F$. 
Having opted for meadows (a meta-design decision) the next question is: which meadows (a design decision). Here are four options:

- $0^{-1} \uparrow$. The value of the expression $0^{-1}$ is undefined, inverse is a partial function.
- $0^{-1} = a$. The value of the expression $0^{-1}$ equals some new constant $a$ which serves as an additional (new) value to data types and represents some form of error).
- $0^{-1} = 17$. Some value is chosen “randomly” as a value for $0^{-1}$.
- $0^{-1} = 1$. 1 is chosen as “meaningful” value for $0^{-1}$, expecting this choice to bring some nice equations.
- $0^{-1} = 0$. 0 is chosen as a value for $0^{-1}$, expecting this to be somewhat better than choosing 1 (or 17).
Forget the options $0^{-1} = 17$ and $0^{-1} = 1$ in favor of $0^{-1} = 0$ because the latter leads to nicer equations and to a better meta-theory. Three options remain.

- $0^{-1} \uparrow$. Inverse of zero undefined: partial meadow.
- $0^{-1} = a$. Error algebra architecture: common meadow.
- $0^{-1} = 0$. Zero-totalized inverse: (zero-totalized) meadow.

A meadow has by default a zero-totalized inverse. A common meadow is not a meadow. A partial meadow is not a meadow. Compare with:

(i) a skew field is not a field,
(ii) a multi-set is not a set,
(iii) a former employee is not an employee.
Axioms for commutative rings and axioms for fields

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

Axioms for fields: those above + the following two

\[0 \neq 1\]
\[x \neq 0 \rightarrow \exists y \ x \cdot y = 1\]
Each field $F$ (with domain $\| F \|$) can be expanded with a function $(..)^{-1}$ as follows:

(i) if $x \cdot y = 1$ then $x^{-1} = y$ (applies for all $x \neq 0$).

(ii) for $0^{-1}$ there are three options (partial meadow, common meadow, meadow)

- $0^{-1}$ is undefined,
- $0^{-1} = a$
  
  $a$ denoting a new element (say $a$) added to $\| F \|$ thus obtaining domain $\| F \| \cup \{a\}$ with $a = \| a \|$. Functions must be extended to $a$ as well, for instance: $a^{-1} = a$, $-a = a$, and for $s$ in $\| F \| \cup \{a\}$: $s + a = a + s = s \cdot a = a \cdot s = a$.
- $0^{-1} = 0$
Partial meadows

Uniform notations: for field $F$, $F'$ denotes its expansion to a partial meadows. $F'_2, F'_3, \ldots, F'_p, \ldots, Q', R', \text{ and } C'$.

Partial meadows have advantages:

- close to the “usual view” (standard intuition),
- easy to imagine.

and disadvantages:

- Logics and proof rules must be selected from a large variety of options.
- No obvious logic for this case stands out as being most plausible.
- Corresponding logic and proof rules are hard to design and tend to be counterintuitive,
Consider the partial meadow $\mathbb{Q}'$. This is an obvious candidate “carrier” for school mathematics. Here are some issues:

1. Is the truth value of $0^{-1} = 1$ “true” (very implausible), “false” (plausible) or “undefined”, a third truth value (most plausible).

2. Suppose we consider $0^{-1} = 1$ to be “false”, what about $0^{-1} = 0^{-1}$ (if it is “true”, what about $0^{-1} = 0^{-1} + 1$).

   - Non-classical logics are hard to understand in full detail (much more difficult than ordinary school mathematics is supposed to be).
   - Around 1980 it took the literature on partial abstract data types some 20 papers to get the details right.
   - And even then there are many options: is $0^{-1} = 1 \lor 1 = 1 + 0$ true? Both “true” and “undefined” and are defensible. Is $0^{-1} = 1 \lor 1 = 1 + 0$ the same as $1 = 1 + 0 \lor 0^{-1} = 1$. Again both “true” and “undefined” and are defensible.
Partial meadows: least attractive option

This is a matter of taste but currently I consider partial meadows to be the least attractive option because:

(i) working out any 3-valued logic on top of a partial meadows quite complicated, and

(ii) only the short-circuit logics (asymmetric connectives)

1 = 1 + 0 \lor 0^{-1} = 1 \text{ must be “true”, and}

connectives must be read from left to right, so that

0^{-1} = 1 \lor 1 = 1 + 0 \text{ is “undefined”,}

seem to take ordinary intuitions correctly into account thus adding one more level of complexity.
Common meadows

This is the most common approach in abstract data types:

- Avoid partial functions (and the corresponding complications of logic design) by making functions total, represent the value of an “undefined value” by a new (additional) and artificial object $a$ different from all know elements of a domain. I will use $a$ as the name for that object in the case of meadows, in general one uses error or not-ok or ⊥.

- Application of functions to $a$ produces $a$ ($a$ denotes an error element with error propagation convention).

- This may turn out to be optimal in the case of meadows, I don’t know.

- However: the mathematics of it is not straight forward at all. For instance, instead of $0 \cdot x = 0$ we find that $0 \cdot a = a$. And instead of $x + (-x) = 0$ we find $x + (-x) = 0 \cdot x$.

- We have axioms (equations) but (at this moment) we have no theory about those equations. That may change.

Jan Bergstra (STCS)
Equations for common meadows ($0^{-1} = a$)

This is guesswork! May be redundant, may be seriously incomplete. 7 or 8 of these equations are “foreign” to known mathematics/algebra.

\[
egin{align*}
(x + y) + z &= x + (y + z) \\
x + y &= y + x \\
x + 0 &= x \\
x + (-x) &= 0 \cdot x \\
0 \cdot 0 &= 0 \\
-(x) &= x \\
0 \cdot x \cdot y &= 0 \cdot x + 0 \cdot y \\
(x \cdot y) \cdot z &= x \cdot (y \cdot z) \\
x \cdot y &= y \cdot x \\
x \cdot (y + z) &= x \cdot y + x \cdot z \\
(x^{-1})^{-1} &= x + 0 \cdot x^{-1} \\
x \cdot x^{-1} &= 1 + 0 \cdot x^{-1} \\
(x \cdot y)^{-1} &= x^{-1} \cdot y^{-1} \\
0^{-1} &= a \\
x + a &= a \\
x \cdot a &= a \\
a^{-1} &= a
\end{align*}
\]
Meadows: $0^{-1} = 0$

Third option: $0^{-1} = 0$. This brings some nice equations, and known mathematics (von Neumann regular rings).

\[
(x^{-1})^{-1} = x \\
x \cdot (x \cdot x^{-1}) = x
\]
Md: equations for meadows

\[(x + y) + z = x + (y + z)\]
\[x + y = y + x\]
\[x + 0 = x\]
\[x + (-x) = 0\]
\[(x \cdot y) \cdot z = x \cdot (y \cdot z)\]
\[x \cdot y = y \cdot x\]
\[1 \cdot x = x\]
\[x \cdot (y + z) = x \cdot y + x \cdot z\]
\[(x^{-1})^{-1} = x\]
\[x \cdot (x \cdot x^{-1}) = x\]
The meadow of rational numbers $\mathbb{Q}_0$

$\mathbb{Q}_0$ is the zero-totalized version of the rational numbers.

$\mathbb{Q}_0$ satisfies: equation L

$$\frac{1 + x^2 + y^2}{1 + x^2 + y^2} = 1.$$ 

Md+L constitutes an initial algebra specification of $\mathbb{Q}_0$. 
$\mathbb{R}_0$ is the zero-totalized version of the field of real numbers.

(i) $\mathbb{R}_0$ satisfies: equation $L_n$ for all $n$.

$$\frac{1 + x_1^2 + x_2^2 + \ldots + x_n^2}{1 + x_1^2 + x_2^2 + \ldots x_n^2} = 1.$$ 

(ii) $\text{Md}+L_n \ (n \in \mathbb{N})$ proves all equations true in the meadow $\mathbb{R}_0$. 
$\mathbb{R}_0$ is the zero-totalized version of the field of real numbers.

(i) $\mathbb{C}_0$ satisfies equation $K_n$ for all $n$:

$$\frac{1 + \ldots + 1}{1 + \ldots + 1} = 1. \quad (n \text{ times } 1)$$

(ii) $\text{Md}+K_n \ (n \in \mathbb{N})$ proves all equations true in the meadow $\mathbb{C}_0$. 
Major open problem about $\mathbb{Q}_0$

QUESTION
Is there a reasonable (effectively enumerable) set of equations from which all equations true in $\mathbb{Q}_0$ can be derived.

REMARK
Equivalent with a longstanding issue: Hilbert’s 10th problem for the rational numbers. (Effective solvability of rational Diophantine equations.)
Conclusions on meadows

- The equational theory of meadows is quite ordinary in spite of its somehow controversial design.
- I expect a significant (and rewarding) simplification of exposition if school mathematics is based on meadows rather than on fields.
- It opens a straightforward path to the use of logic and term-rewriting techniques for arithmetical algebra, which is now obstructed by the impenetrable conceptual complexity of the logic of partial functions.