

On meadows and fields

Jan Bergstra

Section Theoretical Software Engineering
Informatics Institute
Universiteit van Amsterdam
janb@science.uva.nl

BMC 4-19-2007

Based on work done in cooperation with

J.V. Tucker (Computer Science, Swansea)

&

Y. Hirschfeld (Mathematics, Tel Aviv)

ABSTRACT DATA TYPES

- Data type = Data + Operations
- Subject dates back to the mid 60's.
Hupbach, Kaphengst, Reichel:
Robotron Dresden, then East Germany.
- 70's: several USA based individuals and groups, notably Guttag, Liskov & Zilles, ADJ (Goguen, Thatcher, Wagner, Wright), Kamin
- Move back to Europe: Ehrich, Ehrig, Maijster, Klaëren and many more (including JVT and JAB).

Data types are algebras

- Point of departure: Abstract data type = pair (Σ, A) of signature Σ and corresponding algebra A .
- Signature Σ : list of sort names, constant names and function names (no relations)

Data types are algebras

- Point of departure: Abstract data type = pair (Σ, A) of signature Σ and corresponding algebra A .
- Signature Σ : list of sort names, constant names and function names (no relations)
- Many-sorted algebra: more than one sort in the signature.
- Algebra A is **minimal** \leftrightarrow all objects of all sorts of A can be found as the interpretation of a closed expression (term) over the signature.

Examples of Abstract Data Types I

Signature: Σ_{nat} can be denoted by

Syntax

signature Σ_{nat}

sorts nat

operations

0 : $\rightarrow nat$;

1 : $\rightarrow nat$;

$+$: $nat \times nat \rightarrow nat$;

Examples of Abstract Data Types I

Signature: Σ_{nat} can be denoted by

Syntax

signature Σ_{nat}

sorts nat

operations

0 : $\rightarrow nat$;

1 : $\rightarrow nat$;

$+$: $nat \times nat \rightarrow nat$;

The data type \mathbb{N} with constants 0 and 1 and operator $+$ consists of a countable number of objects each the interpretation of an expression of the form $1 + \dots + 1$.

Examples of Abstract Data Types I

Signature: Σ_{nat} can be denoted by

Syntax

signature Σ_{nat}

sorts nat

operations

$0: \rightarrow nat;$

$1: \rightarrow nat;$

$+: nat \times nat \rightarrow nat;$

The data type \mathbb{N} with constants 0 and 1 and operator $+$ consists of a countable number of objects each the interpretation of an expression of the form $1 + \dots + 1$.

Another datatype with the **same signature** is $\mathbb{Z}/(5 \cdot \mathbb{Z})$ ignoring subtraction.

Examples of Abstract Data Types II

Signature: Σ_{intag} (*ag* for additive group) can be denoted by

Syntax

signature Σ_{intag}

sorts *int*

operations

0: $\rightarrow int$;

1: $\rightarrow int$;

+: $int \times int \rightarrow int$;

-: $int \rightarrow int$;

Conventional abbreviation: $x - y = x + (-y)$.

Data types: \mathbb{Z} and $\mathbb{Z}/(k \cdot \mathbb{Z})$ for $k \in \mathbb{N}$.

More precision may be needed for distinguishing sort name in syntax (signature) and semantics (data type). For instance:

INT = (*int*, 0, 1, *plus*, *minus*) where $int = |\mathbb{Z}|$.

Syntax

signature Σ_{int}

sorts int

operations

$0: \rightarrow int;$

$1: \rightarrow int;$

$+: int \times int \rightarrow int;$

$-: int \rightarrow int;$

$\cdot: int \times int \rightarrow int;$

Syntax

signature Σ_{int}

sorts int

operations

$0: \rightarrow int;$

$1: \rightarrow int;$

$+: int \times int \rightarrow int;$

$-: int \rightarrow int;$

$\cdot: int \times int \rightarrow int;$

Data types: \mathbb{Z} and $\mathbb{Z}/(k \cdot \mathbb{Z})$ for $k \in \mathbb{N}$.

Syntax

signature Σ_{int}

sorts int

operations

$0: \rightarrow int;$

$1: \rightarrow int;$

$+: int \times int \rightarrow int;$

$-: int \rightarrow int;$

$\cdot: int \times int \rightarrow int;$

Data types: \mathbb{Z} and $\mathbb{Z}/(k \cdot \mathbb{Z})$ for $k \in \mathbb{N}$.

Notice: $\Sigma_{int} = \Sigma_{intag} \oplus [\cdot: int \times int \rightarrow int]$

Software Engineering: algebra of signatures (**module algebra**), no real issue at the scale of this talk.

Renamed signature:

Syntax

signature Σ_{CR}

sorts *ring*

operations

0: \rightarrow *ring*;

1: \rightarrow *ring*;

+: *ring* \times *ring* \rightarrow *ring*;

-: *ring* \rightarrow *ring*;

·: *ring* \times *ring* \rightarrow *ring*;

Renamed signature:

Syntax

signature Σ_{CR}

sorts *ring*

operations

0: $\rightarrow ring$;

1: $\rightarrow ring$;

+: $ring \times ring \rightarrow ring$;

-: $ring \rightarrow ring$;

·: $ring \times ring \rightarrow ring$;

Advantage of signature: (i) precise syntax available, (ii) support for distinction between syntax and semantics

Renamed signature:

Syntax

signature Σ_{CR}

sorts *ring*

operations

0: $\rightarrow ring$;

1: $\rightarrow ring$;

+: $ring \times ring \rightarrow ring$;

-: $ring \rightarrow ring$;

·: $ring \times ring \rightarrow ring$;

Advantage of signature: (i) precise syntax available, (ii) support for distinction between syntax and semantics

ADTs = models = total Σ_{CR} -algebras (includes all commutative rings with unit).

equations CR (Commutative Ring)

$$(x + y) + z = x + (y + z)$$

$$x + y = y + x$$

$$x + 0 = x$$

$$x + (-x) = 0$$

$$(x \cdot y) \cdot z = x \cdot (y \cdot z)$$

$$x \cdot y = y \cdot x$$

$$x \cdot 1 = x$$

$$x \cdot (y + z) = x \cdot y + x \cdot z$$

Let A be a Σ -algebra for some signature:

A is **minimal** if it has no proper subalgebras (that is, each element of each sort is the interpretation of some closed term)

A is computable if

Theorem (Bergstra & Tucker 1979....) *Each minimal and computable algebra A is the initial algebra $I(\Sigma, E)$ of some finite equational specification (Σ, E) which may perhaps make use of so-called auxiliary or hidden functions.*

(Σ, E) specifies $A \Leftrightarrow A \cong I(\Sigma, E)$

Let A be a Σ -algebra for some signature:

A is **minimal** if it has no proper subalgebras (that is, each element of each sort is the interpretation of some closed term)

A is computable if

Theorem (Bergstra & Tucker 1979....) *Each minimal and computable algebra A is the initial algebra $I(\Sigma, E)$ of some finite equational specification (Σ, E) which may perhaps make use of so-called auxiliary or hidden functions.*

(Σ, E) specifies $A \Leftrightarrow A \cong I(\Sigma, E)$

Equational specifications are known in group theory as *presentations*.

Initial algebra construction

$T(\Sigma)$: set of closed terms over Σ .

$$I(\Sigma, E) \cong T(\Sigma) / \equiv_E$$

\vdash : formal derivability in predicate logic.

$$t \equiv_E r \Leftrightarrow E \vdash t = r$$

The initial algebra is a model of its specification:

$$I(\Sigma, E) \models E$$

Let R be a commutative ring then:

$$R \models CR$$

Let R be a commutative ring then:

$$R \models CR$$

that is, for each equation $e_1 = e_2 \in CR$:

$$R \models e_1 = e_2$$

Let R be a commutative ring then:

$$R \models CR$$

that is, for each equation $e_1 = e_2 \in CR$:

$$R \models e_1 = e_2$$

R cannot be empty but $0 = 1$ may hold.

remarks on (Σ_{CR}, CR)

Let R be a commutative ring then:

$$R \models CR$$

that is, for each equation $e_1 = e_2 \in CR$:

$$R \models e_1 = e_2$$

R cannot be empty but $0 = 1$ may hold.

Initial Algebra of the ADT specification (Σ_{CR}, CR) :

$$I(\Sigma_{CR}, CR) \cong \mathbb{Z}$$

remarks on (Σ_{CR}, CR)

Let R be a commutative ring then:

$$R \models CR$$

that is, for each equation $e_1 = e_2 \in CR$:

$$R \models e_1 = e_2$$

R cannot be empty but $0 = 1$ may hold.

Initial Algebra of the ADT specification (Σ_{CR}, CR) :

$$I(\Sigma_{CR}, CR) \cong \mathbb{Z}$$

$I(\Sigma, E)$ is initial in the category of Σ algebras that satisfy E .

Algebraic specifications I

Specification problem for a **given** data type $A \in Alg(\Sigma)$.

Find equations E over signature Σ such that A is isomorphic to the free term algebra of closed Σ -terms modulo the smallest congruence generated by the equations E ($A \cong I(\Sigma, E)$).

If that is not possible, add one or more additional operations (auxiliary functions, hidden functions) to the signature Σ , obtaining Σ' and expand A to a $A' \in Alg(\Sigma')$ such that a specification (Σ', E') can be found which specifies A' . (An expansion has the same domains, only more functions).

Algebraic specifications II

Alternatively: use a **more expressive specification language**, for instance conditional equations for which the initial algebra construction works equally well. There is a limit to this, for instance if **inequality** or **disjunction** is included in the specification notation the initial algebra construction fails.

Specification problem for a given data type $A \in Alg(\Sigma)$.

EAS, elementary algebraic specifications (Σ, E) :

- 1 all sorts are non-empty and all functions are total. REASON: that yields by far the simplest equational logic.
- 2 equations or conditional equations.
- 3 finite number of equations (conditional equations)
- 4 auxiliary functions.

Many non-elementary features have been developed in recent years to find more expressive specification formalisms. I will leave those aside.

Algebraic specifications IV

Measuring the quality of an EAS.

Non-technical qualities:

- 1 readable and informative.
- 2 simplicity relative to alternative descriptions of the same algebra (for instance based on algorithms for all functions and a representation of the domains in some known data structures).
- 3 'natural' auxiliary functions (if any are used).

Technical virtues (TRS = term rewriting system):

- 1 (Σ, E) provides a **confluent** TRS or a **terminating** TRS or both (**complete** TRS).
- 2 as a TRS, (Σ, E) has nice properties modulo AC (associativity & commutativity).

Number algebras as data types I

The natural numbers with 0, 1 and addition have this EAS (elementary algebraic specification):

(Σ_{nat}, E_{nat}) with

$$E_{nat} = \{x + 0 = x, x + y = y + x, x + (y + z) = (x + y) + z\}$$

Extending the signature with an inverse for addition (obtaining Σ_{intag}) the integers are an additive group specified by:

$(\Sigma_{intag}, E_{intag})$ with

$$E_{intag} = E_{int} \cup \{x + (-x) = 0\}$$

Number algebras as data types II

Add a function symbol \cdot for multiplication (obtaining Σ_{CR}) the integers constitute a ring \mathbb{Z} specified by:

(Σ_{CR}, E_{CR}) with
 $E_{CR} = CR$

Now consider the rational numbers \mathbb{Q} as a Σ_{CR} algebra: indeed $\mathbb{Q} \in Alg(\Sigma_{CR})$, but \mathbb{Q} is not minimal.

Each EAS for \mathbb{Q} needs at least one additional operation for generating the whole domain of \mathbb{Q} .

Larry Moss (2001) adds a unary function (to Σ_{CR} , expanding \mathbb{Q}) and obtains a remarkable (though unreadable) equational specification.

The rational numbers as an abstract data type

Extend the signature Σ_{CR} with a function symbol $^{-1}$ for inverse (unary division).

EAS requires that all functions are total. Many options exist, Hodges, and more systematically Harrison (1998) use zero totalization:
 $0^{-1} = 0$.

We will follow that line to its conclusion in the matter of algebraic specifications for fields.

The signature of meadows Σ_m

Add a unary operator for denoting an inverse to Σ_{CR} .

Syntax

signature Σ_m

sorts *ring*

operations

$0: \rightarrow \textit{ring};$

$1: \rightarrow \textit{ring};$

$+: \textit{ring} \times \textit{ring} \rightarrow \textit{ring};$

$-: \textit{ring} \rightarrow \textit{ring};$

$\cdot: \textit{ring} \times \textit{ring} \rightarrow \textit{ring};$

$^{-1}: \textit{ring} \rightarrow \textit{ring}$

$^{-1}$ is supposed to be a total function.

DEFINITION: a meadow is Σ_m -structure that satisfies *CR* (that is a commutative ring with unit) with $^{-1}$ obeying the following laws:

equations CR (Commutative Ring)

$$\begin{aligned}x \cdot (x \cdot x^{-1}) &= x \\(x^{-1})^{-1} &= x\end{aligned}$$

These equations are named; *Ril* (*restricted inverse law*) and *Ref* (*reflection*).

We will write: $Md = CR \cup \{Rel, ref\}$

Many examples of meadows can be constructed.

Examples of meadows I: zero totalized fields

Let F be a field in the signature Σ_{CR} of rings. Expand F with a zero totalized division operator:

$$x^{-1} = y \text{ such that } x \cdot y = 1 \text{ if } x \neq 0, 0 \text{ otherwise}$$

Conventional abbreviation: $\frac{x}{y} = x \cdot y^{-1}$.

Thus in conventional notation: $\frac{1}{0} = 0$ from which using CR one obtains $\frac{x}{0} = 0$ for all x .

The new field is denoted F_0 to indicate that its division operator has been made zero totalized.

Examples of meadows II: products of zero totalized fields

Note that the meadows are defined as a model class of equations:
following universal algebra meadows are closed under:

taking subalgebras,
direct products, and
homomorphic images.

Products of zero totalized fields are meadows, for instance:

\mathbb{R}_0^n for $n \in \mathbb{N}$.

Examples of meadows III: $(\mathbb{Z}/k \cdot \mathbb{Z})$ with k squarefree

$k \in \mathbb{N}$ is squarefree iff it is a product of different primes. Consider

$$E_k = Md \cup \{\underline{k} = 0\}$$

with $\underline{0} = 0$, $\underline{k+1} = \underline{k} + 1$. Now:

THEOREM: $I(\Sigma_m, E_k)$ is a meadow with k elements.

Note: $I(\Sigma_m, E_k)$ is a product of zero totalized prime fields.

equations SIP (Strong Inverse Properties, SIP1, SIP2 and SIP3)

$$\begin{aligned}(-x)^{-1} &= -(x^{-1}) \\(x \cdot y)^{-1} &= x^{-1} \cdot y^{-1} \\(x^{-1})^{-1} &= x\end{aligned}$$

Note SIP3 is *Ril*. Recall $Md = CR \cup \{Rel, Ril\}$. Now

THEOREM: SIP1 and SIP2 follow from Md (i.e., $Md \vdash \text{SIP}_i$).

THEOREM: $Md \vdash 0^{-1} = 0$.

Another consequence is: $x \cdot y = 1 \rightarrow y = x^{-1}$.

Algebraic specification of the zero totalized rationals

\mathbb{Q}_0 is a minimal Σ_m -algebra expanding \mathbb{Q} . We need an abbreviation $Z()$ and one more equation L (L for Lagrange)

equations Z and L

$$\begin{aligned}Z(x) &= 1 - \frac{x}{x} \\Z(1 + x^2 + y^2 + z^2 + u^2) &= 0\end{aligned}$$

THEOREM (Bergstra & Tucker 2007, J. ACM):

$$\mathbb{Q}_0 \cong I(\Sigma_m, CR \cup SIP \cup L)$$

Algebraic specification of the zero totalized rationals

\mathbb{Q}_0 is a minimal Σ_m -algebra expanding \mathbb{Q} . We need an abbreviation $Z()$ and one more equation L (L for Lagrange)

equations Z and L

$$\begin{aligned}Z(x) &= 1 - \frac{x}{x} \\Z(1 + x^2 + y^2 + z^2 + u^2) &= 0\end{aligned}$$

THEOREM (Bergstra & Tucker 2007, J. ACM):

$$\mathbb{Q}_0 \cong I(\Sigma_m, CR \cup SIP \cup L)$$

Equivalently: $\mathbb{Q}_0 \cong I(\Sigma_m, Md \cup L)$

$\mathbb{C}\mathbb{Q}_0$ is a minimal $\Sigma_m \cup [i : \rightarrow \text{ring}]$ -algebra extending \mathbb{Q}_0 .

It might have an equational EAS without auxiliary functions. (Open question)

Using a unary auxiliary operator cc for the complex conjugate an equational EAS can be given (quite similar to the specification for \mathbb{Q}_0).

Rational functions in a single variable

This is what we obtain by adding a single constant X to the signature Σ_m (obtaining $\Sigma_{m,X}$). Because no specific equations are assumed for X we get an algebra isomorphic to the result of adjunction of a transcendental number (e.g. π) to \mathbb{Q}_0 (obtaining $\mathbb{Q}_0[\pi]$).

Rational functions in a single variable

This is what we obtain by adding a single constant X to the signature Σ_m (obtaining $\Sigma_{m,X}$). Because no specific equations are assumed for X we get an algebra isomorphic to the result of adjunction of a transcendental number (e.g. π) to \mathbb{Q}_0 (obtaining $\mathbb{Q}_0[\pi]$).

First, define $N(x) = 1 - Z(x) = x \cdot x^{-1}$, now add d (for degree) to $\Sigma_{m,X}$ (obtaining $\Sigma_{m,X,d}$).

Rational functions in a single variable

This is what we obtain by adding a single constant X to the signature Σ_m (obtaining $\Sigma_{m,X}$). Because no specific equations are assumed for X we get an algebra isomorphic to the result of adjunction of a transcendental number (e.g. π) to \mathbb{Q}_0 (obtaining $\mathbb{Q}_0[\pi]$).

First, define $N(x) = 1 - Z(x) = x \cdot x^{-1}$, now add d (for degree) to $\Sigma_{m,X}$ (obtaining $\Sigma_{m,X,d}$).

Expand the algebra $\mathbb{Q}_0[X]$ with a degree function d . d computes the degree of a closed term over $\Sigma_{F,i,X}$ viewed as a rational function (= a quotient of polynomials) over \mathbb{Q}_0 (obtaining the algebra $\mathbb{Q}_0[X, d]$).

Rational functions in a single variable

This is what we obtain by adding a single constant X to the signature Σ_m (obtaining $\Sigma_{m,X}$). Because no specific equations are assumed for X we get an algebra isomorphic to the result of adjunction of a transcendental number (e.g. π) to \mathbb{Q}_0 (obtaining $\mathbb{Q}_0[\pi]$).

First, define $N(x) = 1 - Z(x) = x \cdot x^{-1}$, now add d (for degree) to $\Sigma_{m,X}$ (obtaining $\Sigma_{m,X,d}$).

Expand the algebra $\mathbb{Q}_0[X]$ with a degree function d . d computes the degree of a closed term over $\Sigma_{F,i,X}$ viewed as a rational function (= a quotient of polynomials) over \mathbb{Q}_0 (obtaining the algebra $\mathbb{Q}_0[X, d]$).

Then consider the equations DG over the signature $\Sigma_{m,X,d}$:

equations DG (degree)

$$d(0) = 0$$

$$d(1) = 0$$

$$d(X) = 1$$

$$d(X + 1) = 1$$

$$d(-x) - d(x) = 0$$

$$d(x^{-1}) + d(x) = 0$$

$$d(d(x)) = 0$$

$$N(y) \cdot d(x) + N(x) \cdot d(y) - d(x \cdot y) = 0$$

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

$$N(d(x)) \cdot Z(x) = 0$$

THEOREM. $I(\Sigma_{m,X,d}, Md + L + DS) \cong \mathbb{Q}_0[X, d]$

QUESTION: Find a specification for $\mathbb{Q}_0[X, Y, d]$ (open).

REPRESENTATION THEOREM: every non-trivial meadow is a substructure of a product of zero-totalized fields.

COROLLARY: the equations true of all zero totalized fields coincide with the equations true of all meadows.

in other words: the equational theory of totalized fields has a finite basis (e.g., Md).

The proof of the representation theorem uses Zorn's lemma (for defining a maximal ideal).

Is AC needed for the corollary? (Hard to imagine.)

Connection with term rewriting

ADEQUACY THEOREM (Bergstra & Tucker 1995): Any computable data type can be provided with a finite equational initial algebra specification making use of hidden functions, in such a way that an **orthogonal term rewriting system** is obtained (hence confluent) which is moreover **strongly normalizing**.

Thus considering \mathbb{Q}_0 (given that it is a computable data type), its specification is informative (compared with more general theory) because NO hidden functions are used. For $\mathbb{Q}_0[X]$ the point is that only ONE unary hidden function d is used in the specification of $\mathbb{Q}_0[X, d]$.

QUESTION: Can \mathbb{Q}_0 and $\mathbb{Q}_0[X, d]$ be specified using orthogonal term rewriting systems?

QUESTION: Can $\mathbb{Q}_0[X]$ be specified without the use of an auxiliary function?

And finally what is E in EAS?

EAS :

- 1 specification of a single minimal data type (up to isomorphism)
- 2 many sorted signature
- 3 constants and total functions, no relations
- 4 equations and conditional equations
- 5 auxiliary functions (no auxiliary sorts)
- 6 initial algebra semantics

Non-elementary features:

- 1 partial functions
- 2 subsorts
- 3 higher types
- 4 relations
- 5 general first order specifications
- 6 bound variables
- 7 infinite signatures and equation systems

Thank you for your attention!