BINARY STRUCTURES IN PROGRAM TRANSFORMATIONS

H. Zantema

RUU-CS-88-24
July 1988

Rijksuniversiteit Utrecht

Vakgroep informatica

Padualaan 14 3584 CH Utrecht
Corr. adres: Postbus 80.089, 3508 TB Utrecht
Telefoon 030-531454
The Netherlands

BINARY STRUCTURES IN PROGRAM TRANSFORMATIONS

H. Zantema

RUU-CS-88-24
July 1988

Rijksuniversiteit Utrecht

Vakgroep informatica

Padualaan 14 3584 CH Utrecht
Corr. adres: Postbus 80.089, 3508 TB Utrecht
Telefoon 030-531454
The Netherlands

BINARY STRUCTURES IN PROGRAM TRANSFORMATIONS

H. Zantema

Technical Report RUU-CS-88-24
July 1988

Department of Computer Science
University of Utrecht
P.O.Box 80.089, 3508 TB Utrecht
The Netherlands

Binary structures in program transformations

H. Zantema

Abstract

The initial algebra approach is used to give a formal definition of a binary structure as it appears in the Bird–Meertens formalism of algorithmic program transformation. Both transformation rules and conditions for adding a unit element are derived from this definition.

Contents

1 Introduction

In various papers on formalizing program transformations ([1,2]) some data structures are described as a *binary structure* over a *domain*. A typical example is the binary structure *finite sequence* over the domain *integers*. Essential for such a structure is the existence of an embedding (the singleton-constructor) of the domain into the structure, and a binary operator on the structure. This binary operator has to satisfy some prescribed laws. In the example of finite sequences over the integers the operator is the concatenation of finite sequences, while *associativity* is the only prescribed law.

Often such a structure is introduced informally, without an exact definition. This paper is an attempt to present a binary structure over any given domain (being a set) and any set of laws (being universally quantified equations) in a more formal way. The definition is given by a universal property, from which various properties are derived, like the homomorphism lemma and the promotion lemmas from [1,2]. It can also be used for defining a natural relation on sets of laws. Another topic is under which conditions a unit element and the filter operator can be introduced in a consistent way. For example, if the set of laws does not satisfy certain conditions, then after adding an abstract unit element the laws will not hold any more.

The universal property appears to be very powerful. The strategy of this paper is to prove laws and lemmas in [1,2] from this single universal property. Each of these laws represents a transformation rule on functional programs; after [1] the calculus of these transformation rules is called *algorithmics*. On a metalevel a remarkable property holds: if two algorithmic expressions have the same type then they are equal (and hence can be used as a tranformation rule). This fact can be understood by the universal property.

Given some domain and some set of laws, the universal property does not say anything about the *existence* of such a binary structure. The existence is shown by the outline of a construction in section 8. In this construction, a congruence relation is constructed and the quotient of the term algebra is taken. For details we refer to [4]. A reader convinced of the existence right away, can skip section 8, and does not need to worry about the shape of the construction.

In terms of the initial algebra approach as discussed in [3,4], one can take a signature Σ with one sort, where the operations consist of a constant for each element of the domain, and one binary operation. Then our binary structure is the initial algebra in the variety of Σ-algebras of the given set of laws, that is the class consisting of all Σ-algebras satisfying these laws.

However, we do not assume that everyone is familiar with this. For reading this paper, knowledge of universal algebra or category theory is not required. On the other hand, the simple signature we restrict to reflects a substantial part of the behaviour of general signatures. A lot of our propositions can be made more general simply by adding some notational decorations.

The basic notions are introduced in section 2. Up to minor details, sections 3, 4, 5 and 8 depend only on section 2, while section 6 depends on sections 2 and 4. In section 7 a rather unfamiliar example is treated to which the results can be applied.

2 The universal property

First we want to formalize the notion of a *law* on a binary operator, like the commutative law and the associative law. Sometimes laws are called equations in the literature.

Definition 1 *A* law *on n variables $x_1, x_2, \cdots, x_n$ is a pair of terms, where a* term *t is defined recursively by the syntax rule*

$$t ::= x_1 \mid x_2 \mid \cdots \mid x_n \mid (t \oplus t).$$

For example, the laws

$$\begin{array}{l}(x_1, (x_1 \oplus x_1)),\\ ((x_1 \oplus x_2), (x_2 \oplus x_1)), \text{and}\\ (((x_1 \oplus x_2) \oplus x_3), (x_1 \oplus (x_2 \oplus x_3)))\end{array}$$

are the *idempotent, commutative* and *associative* laws respectively. Note that this definition is independent of the environment where the law is expected to hold. The correspondence between laws and their environments is given in the next definition.

Definition 2 *Let A be some set and $\odot : A \times A \to A$ a binary operator on A. For elements $x_1, x_2, \cdots, x_n$ of A, and a term t on $x_1, x_2, \cdots, x_n$, we denote by t_A the element of A obtained from t by replacing all symbols $\oplus$ by $\odot$.*

We say that a set of laws L holds *in $(A, \odot)$ (or $(A, \odot)$* satisfies *L) if*

$$\forall (t, t') \in L : \ \forall x_1, x_2, \cdots, x_n \in A : \ t_A = t'_A.$$

Let an arbitrary set D be given and also any set L of laws. One purpose of this paper is to present a binary structure $S_{D,L}$: the *smallest* set with a binary operation, which

- allows embedding of D,
- satisfies L, and
- has no *confusion*, i. e. no essentially larger set of laws than L holds.

On the other hand, it is also the *greatest* set with a binary operation which

- allows embedding of D,
- satisfies L, and

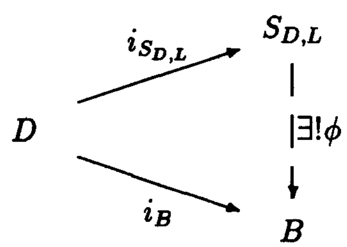
- has no *junk*, i. e. each element can be expressed as a term with variables from D.

These ideas are made more precise in the following definition, which is central to the whole paper. As nothing has been said about existence and uniqueness of such a smallest or greatest set with some desired properties, we can not yet speak about *the* binary structure $S_{D,L}$ but only about the $S_{D,L}$-property.

Definition 3 (universal property) *Let D be any set and let L be a set of laws. Let A be a set, $i_A : D \to A$ and $\odot : A \times A \to A$.*

We say that $(A, i_A, \odot)$ has the $S_{D,L}$-property *if*

- *L holds in $(A, \odot)$, and*
- *if B is a set, $i_B : D \to B$ and $\otimes : B \times B \to B$, such that L holds in $(B, \otimes)$, then there is* exactly *one $\phi : A \to B$ with*
  - *$\phi \circ i_A = i_B$, and*
  - *$\phi(a \odot a') = \phi(a) \otimes \phi(a')$ for all $a, a' \in A$.*



Intuitively one can say that having 'no junk' corresponds to the existence of at most one such ϕ, while having 'no confusion' corresponds to the existence of at least one such ϕ.

A triple $(A, i_A, \odot)$ with $i_A : D \to A$ and $\odot : A \times A \to A$ is called an *algebra* over D. If it is clear which i_A and $\odot$ are meant, we sometimes speak about the algebra A. A consequence of the definition will be the uniqueness of an algebra $(S_{D,L}, i_{S_{D,L}}, \oplus)$ with the $S_{D,L}$-property up to isomorphism. Before we can state this, we have to define what is meant by isomorphism of algebras.

Definition 4 *Let D be any set and let $\mathcal{A} = (A, i_A, \odot)$ and $\mathcal{B} = (B, i_B, \otimes)$ be algebras over D. A map $\phi : A \to B$ is called a* homomorphism *from $\mathcal{A}$ to $\mathcal{B}$ if*

- *$\phi \circ i_A = i_B$, and*
- *$\phi(a \odot a') = \phi(a) \otimes \phi(a')$ for all $a, a' \in A$.*

Two algebras $\mathcal{A}$ and $\mathcal{B}$ over D are called isomorphic *if there exist homomorphisms $\phi : \mathcal{A} \to \mathcal{B}$ and $\psi : \mathcal{B} \to \mathcal{A}$ such that*

$$\phi \circ \psi = id_B \quad and \quad \psi \circ \phi = id_A.$$

Note that the composition of a homomorphism from $\mathcal{A}$ to $\mathcal{B}$ and a homomorphism from $\mathcal{B}$ to $\mathcal{C}$ is a homomorphism from $\mathcal{A}$ to $\mathcal{C}$.

In terms of algebras and homomorphisms, the universal property can be formulated far shorter:

An algebra $\mathcal{A}$ has the $S_{D,L}$-property if and only if

- *$\mathcal{A}$ satisfies L, and*
- *for each algebra $\mathcal{B}$ satisfying L there is exactly one homomorphism from $\mathcal{A}$ to $\mathcal{B}$.*

We are more interested in algebras up to isomorphism than in algebras themselves. Sometimes we shall even call algebras 'equal' if they are isomorphic.

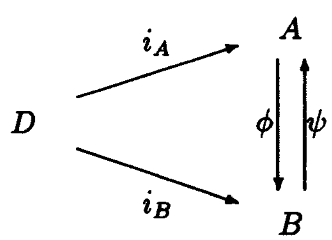
Proposition 1 *Let D be a set and L a set of laws. If two algebras over D both have the $S_{D,L}$-property, then they are isomorphic.*

Proof: Let $\mathcal{A} = (A, i_A, \odot)$ and $\mathcal{B} = (B, i_B, \otimes)$ be algebras over D both having the $S_{D,L}$-property. Then there is exactly one homomorphism

$$\phi : \mathcal{A} \rightarrow \mathcal{B}$$

and exactly one homomorphism

$$\psi : \mathcal{B} \rightarrow \mathcal{A}.$$



Both $\psi \circ \phi$ and id_A are homomorphisms from $\mathcal{A}$ to $\mathcal{A}$. But again using the definition of $S_{D,L}$-property we see that there is exactly one homomorphism from $\mathcal{A}$ to $\mathcal{A}$, so

$$\psi \circ \phi = id_A.$$

In the same way we have

$$\phi \circ \psi = id_B,$$

proving that $\mathcal{A}$ and $\mathcal{B}$ are isomorphic. □

This proof does not use anything particular about algebras. In fact, it is the proof that an initial object in any category is unique up to isomorphism.

In section 8 it is shown by a construction that for each domain D and for each set of laws L an algebra having the $S_{D,L}$-property exists. Since we know that it is unique (up to isomorphism), we may call it *the* algebra $(S_{D,L}, i_{S_{D,L}}, \oplus)$, or for shorthand $S_{D,L}$. This algebra is called the *binary structure* over D with laws L. The map $i_{S_{D,L}}$ is called the *singleton-constructor* and is denoted by $\hat{}$ in [1].

Proposition 2 (no junk) *For each domain D and each set of laws L there is no junk in $S_{D,L}$, i.e., each element of $S_{D,L}$ can be written as a term in which each variable is replaced by an element of the shape $i_{S_{D,L}}(d)$ with $d \in D$.*

Proof: Let $\tilde{S}_{D,L}$ be the set of elements of $S_{D,L}$ that can be written in that way. With the same $i_{S_{D,L}}$ and $\oplus$ this $\tilde{S}_{D,L}$ is an algebra over D. Let

$$\phi : S_{D,L} \rightarrow \tilde{S}_{D,L}$$

be the unique homomorphism, and let

$$i : \tilde{S}_{D,L} \rightarrow S_{D,L}$$

be the inclusion map that maps each element on itself. This inclusion map is a homomorphism. Then both $i \circ \phi$ and the identity map are homomorphisms from $S_{D,L}$ to itself, so they are equal. Hence $i \circ \phi$ is surjective, so i is also surjective, which we had to prove. □

A direct consequence of the fact that $S_{D,L}$ has no junk is the following.

Proposition 3 (induction lemma) *Let D and X be sets and L a set of laws. Let f and g be maps from $S_{D,L}$ to X satisfying the following properties:*

- $f \circ i_{S_{D,L}} = g \circ i_{S_{D,L}}$, *and*
- *if $f(s) = g(s)$ and $f(s') = g(s')$ for $s, s' \in S_{D,L}$, then also $f(s \oplus s') = g(s \oplus s')$.*

Then $f = g$.

It would have been more in the style of this paper to give a proof of this induction lemma directly from the universal property. However, we did not succeed since X is an arbitrary set instead of an algebra satisfying L.

3 A relation on laws

The natural opposite of the proposition stating that $S_{D,L}$ does not contain junk is the property that $S_{D,L}$ does not have confusion. This property will be something like

No essentially larger set of laws than L holds in $S_{D,L}$.

However, we have not yet defined what is meant by 'not essentially larger'. A natural definition of this relation is forced by the no-confusion-property as follows:

Definition 5 *Let D be any set and let L and L' be two sets of laws. We write*

$$L' \prec_D L$$

if L' holds in $S_{D,L}$.

Proposition 4 *Let D be any set and let L and L' be two sets of laws with $L' \subset L$. Then*

$$L' \prec_D L.$$

Proof: Immediate. □

Proposition 5 (projection) *Let D be any set and let L and L' be two sets of laws. Then*

$$L' \prec_D L$$

if and only if there exists a homomorphism

$$\pi : S_{D,L'} \rightarrow S_{D,L}.$$

If such a homomorphism π exists, then it is unique and surjective.

The homomorphism π is called the *projection.*

Proof: Assume

$$L' \prec_D L,$$

then L' holds in $S_{D,L}$. According to the universal property of $S_{D,L'}$ the projection homomorphism exists and is unique.

Conversely, assume that the projection homomorphism π exists. Since $S_{D,L}$ does not contain junk, this homomorphism is surjective. Let

$$(t, t')$$

be an arbitrary law in L', on the variables $x_1, x_2, \cdots, x_n$. Choose these $x_1, x_2, \cdots, x_n$ in $S_{D,L}$ arbitrarily. Next choose $\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_n$ in $S_{D,L'}$ in such a way that

$$\pi(\tilde{x}_i) = x_i \text{ for } i = 1, 2, \cdots, n.$$

Then

$$t_{S_{D,L}} = \pi(t_{S_{D,L'}}) = \pi(t'_{S_{D,L'}}) = t'_{S_{D,L}},$$

where the terms $t_{S_{D,L'}}$ and $t'_{S_{D,L'}}$ are over $\tilde{x}_i$ instead of x_i. We conclude that L' holds in $S_{D,L}$, so

$$L' \prec_D L,$$

which we had to prove. □

For example, if L' contains only associativity, and L contains associativity, commutativity and idempotency, then the projection π maps a finite sequence over D to the set of elements of that finite sequence.

Proposition 6 *Let D be any set. Then $\prec_D$ is a reflexive and transitive relation on the set of laws.*

Proof: Refexivity is immediate from the definition. Assume that

$$L'' \prec_D L' \quad \text{and} \quad L' \prec_D L,$$

according to the last proposition there exist homomorphisms

$$\pi' : S_{D,L''} \to S_{D,L'} \quad \text{and} \quad \pi : S_{D,L'} \to S_{D,L}.$$

Then $\pi \circ \pi' : S_{D,L''} \to S_{D,L}$ is again a homomorphism, so $L'' \prec_D L$, and $\prec_D$ is transitive. □

Define

$$L' \sim_D L \quad \text{if} \quad L \prec_D L' \wedge L' \prec_D L.$$

According to the previous proposition the relation $\sim_D$ is an equivalence relation on the set of sets of laws. Whithout a proof we mention that the equivalence classes of $\sim_D$ form a lattice with the partial order induced by $\prec_D$.

It is an interesting question whether $\prec_D$ depends on D. Since

$$S_{\emptyset,L} = \emptyset$$

we have

$$L' \prec_\emptyset L \quad \text{for each} \quad L, L'.$$

Further one can easily show that

if $\sharp D \leq \sharp D'$ and $L' \prec_{D'} L$, then $L' \prec_D L$.

The relation $\prec_D$ really depends on the number of elements of D, for example let

$$L = \{\text{associativity}\}$$

and

$$L' = \{\text{associativity, commutativity}\}.$$

Then

$$L' \prec_{\{1\}} L$$

and

$$L' \not\prec_{\{1,2\}} L.$$

For each finite D a similar (but more complicated) example can be constructed; this observation is due to E. Lippe.

4 The unit element

In some of the basic tools of algorithmics, in particular the *filter* operator, it is essential to have a *unit element*, sometimes also called *identity element.* First let us give the definition.

Definition 6 *Let $(A, i_A, \odot)$ be an algebra over a domain. An element $u \in A$ is called a* unit element *in the algebra if*

$$u \odot a = a \odot u = a \quad \textit{for all} \quad a \in A.$$

Proposition 7 *An algebra over a domain contains at most one unit element.*

Proof: Assume that both u and u' are unit elements in an algebra $(A, i_A, \odot)$. Then

$$u = u \odot u' = u'.$$

□

The particular algebra $S_{D,L}$ as defined in section 2 will not always contain a unit element. If a unit element is required, a natural way to define the corresponding binary structure — similar to the universal property — is the following.

Definition 7 (extended universal property) *Let D be any set and let $\mathcal{A} = (A, i_A, \odot)$ and $\mathcal{B} = (B, i_B, \otimes)$ be algebras over D, with unit elements u_A and u_B respectively. A map $\phi : A \to B$ is called an* extended homomorphism *from $\mathcal{A}$ to $\mathcal{B}$ if*

- $\phi(u_A) = u_B$, *and*
- $\phi \circ i_A = i_B$, *and*
- $\phi(a \odot a') = \phi(a) \otimes \phi(a')$ *for all* $a, a' \in A$.

An algebra $\mathcal{A}$ over D has the $S^u_{D,L}$-property *if*

- *it has a unit element, and*
- *L holds in $\mathcal{A}$, and*
- *for each algebra $\mathcal{B}$ with a unit element in which L holds, there is exactly one extended homomorphism from $\mathcal{A}$ to $\mathcal{B}$.*

Similar to the non-unit case, this algebra $S^u_{D,L}$ is unique up to (extended) isomorphism. Again we denote its binary operator by $\oplus$.

In this section we introduce two conditions on laws. The first is being *conservative*, for which $S_{D,L}$ does not contain a unit element. The second condition is *unit-closedness*, for which one can construct $S^u_{D,L}$ from $S_{D,L}$ by adding an abstract element u and defining

$$u \oplus s = s \oplus u = s$$

for all $s \in S_{D,L}$, and $u \oplus u = u$.

Definition 8 *A law (t,t') is called* conservative *if the variables occurring in t are exactly the same as the variables occurring in t'.*

For example, the idempotent, commutative and associative laws are all three conservative.

Proposition 8 *Let D be a set with at least two elements and let L be a set of conservative laws. Then $i_{S_{D,L}}$ is an injective map and $S_{D,L}$ does not possess a unit element.*

Proof: Let $\mathcal{P}_D$ be the set of finite non-empty subsets of D. Define

$$i : D \rightarrow \mathcal{P}_D$$

by

$$i(d) = \{d\} \quad \text{for all } d \in D.$$

Then $(\mathcal{P}_D, i, \cup)$ is an algebra over D. Let (t,t') be any conservative law. If the variables in t are replaced by arbitrary elements of $\mathcal{P}_D$, and each occurrence of the binary operator in t is replaced by $\cup$, then the result is the union of the corresponding elements of $\mathcal{P}_D$. If the same is done for t', then the same elements of $\mathcal{P}_D$ are obtained. For each possible choice these elements have the same union, so (t,t') holds in $(\mathcal{P}_D, i, \cup)$. Hence each set of conservative laws, in particular L holds in $(\mathcal{P}_D, i, \cup)$. According to the universal property there exists a homomorphism

$$\phi : (S_{D,L}, i_{S_{D,L}}, \oplus) \rightarrow (\mathcal{P}_D, i, \cup).$$

Clearly i is injective. Since $i = \phi \circ i_{S_{D,L}}$, the same holds for $i_{S_{D,L}}$.

Assume that $S_{D,L}$ contains a unit element u. If $\phi(u)$ contains one element, choose $d \in D$ unequal to that element, else choose d to be an element of $\phi(u)$. In either case we have

$$\{d\} \cup \phi(u) \neq \{d\}.$$

We conclude that

$$\begin{aligned} \{d\} &= \phi(i_{S_{D,L}}(d)) \\ &= \phi(i_{S_{D,L}}(d) \oplus u) \\ &= \phi(i_{S_{D,L}}(d)) \cup \phi(u) \\ &= \{d\} \cup \phi(u) \\ &\neq \{d\}, \end{aligned}$$

which contradicts the assumption. □

Both assumptions in this proposition are essential. If, for example, D contains only one element, and L consists of the associative law together with

$$(x, \underbrace{x \oplus x \oplus \cdots \oplus x}_{n\times}),$$

then

$$\underbrace{x \oplus x \oplus \cdots \oplus x}_{(n-1)\times}$$

is a unit element in $S_{D,L}$. In fact $S_{D,L}$ is the cyclic group of $n-1$ elements.

If, on the other hand, D is arbitrary, but L contains the non-conservative law

$$(x, y),$$

then $S_{D,L}$ contains only one element, which is a unit element.

Let L consist of either the associative and the commutative and the idempotent laws, or of all conservative laws, or of anything between. Then it can be shown that there is also a homomorphism

$$\psi : (\mathcal{P}_D, i, \cup) \rightarrow (S_{D,L}, i_{S_{D,L}}, \oplus),$$

mapping $\{d_1, d_2, \cdots, d_n\}$ to

$$i_{S_{D,L}}(d_1) \oplus i_{S_{D,L}}(d_2) \oplus \cdots \oplus i_{S_{D,L}}(d_n).$$

Then both $\phi \circ \psi$ and $\psi \circ \phi$ are identity maps, so $(\mathcal{P}_D, i, \cup)$ and $(S_{D,L}, i_{S_{D,L}}, \oplus)$ are isomorphic. In the notation of the previous section we conclude that

$$L \prec_D \{\text{associativity, commutativity, idempotency}\}$$

for each set L of conservative laws.

The rest of this section is on formally adding a unit element to $S_{D,L}$. Let D be any set and let L be a set of laws. Define

$$S'_{D,L} = S_{D,L} \cup \{u\}$$

for an abstract element u, and

$$s \oplus u = u \oplus s = s \text{ for all } s \in S_{D,L},$$

and $u \oplus u = u$. The set $S'_{D,L}$ defined in this way is an algebra over D containing a unit element u. Note that this trick of adding a unit element can even be executed if there exists a unit element already. However, if this is done the original unit element is not a unit element any more.

Although $S'_{D,L}$ possesses a unit element and seems to be a good candidate for having the $S^u_{D,L}$-property, the problem is whether this algebra satisfies L. In general, it will not, even if all laws in L are conservative. For example, let L consist of

$$((x \oplus y), ((x \oplus x) \oplus y)).$$

If this law would hold in $S'_{D,L}$ then by taking $y = u$ the idempotent law would hold in $S_{D,L}$, which is not true.

However, we can define some further reasonable restrictions apart from being conservative for which $S'_{D,L}$ will be a proper algebra over D in which L holds, and even will be equal to $S^u_{D,L}$. In the next definition the function ρ_x defined on terms can be considered as a simulation of replacing x by an abstract unit element.

Definition 9 *For a variable x let ρ_x be defined inductively on terms:*

$$\begin{array}{lll} \rho_x(y) & = y & \textit{for all variables } y \textit{ (including } x\textit{),} \\ \rho_x(t \oplus x) & = \rho_x(x \oplus t) \ = \rho_x(t) & \textit{for all terms } t, \textit{ and} \\ \rho_x(t \oplus t') & = \rho_x(t) \oplus \rho_x(t') & \textit{for all terms } t, t', \ t \neq x \neq t'. \end{array}$$

A set L of laws is called unit-closed *if for all variables x and for all $(t, t') \in L$ either*

$$\rho_x(t) = \rho_x(t'), \ \textit{ or } \ (\rho_x(t), \rho_x(t')) \in L.$$

For example, if L consists of either the idempotent, the commutative, or the associative law, or of any combination of them, then L is unit-closed.

If L is unit-closed and L contains any non-conservative law, then it can easily be shown that $S_{D,L}$ contains only one element. So unit-closedness is more restrictive than conservativity in non-trivial cases.

Proposition 9 *Let D be any set and let L be a unit-closed set of conservative laws. Then L holds in $S'_{D,L}$, and*

$$S'_{D,L} = S^u_{D,L},$$

up to extended isomorphism.

Proof: First we prove by induction on the size of laws that L holds in $S'_{D,L}$. Here the size of a law is defined to be the total number of occurring $\oplus$-signs. Since L contains only conservative laws, the only possible law in L of size zero is (x, x) for some variable symbol x, and this law holds in every algebra.

Let (t, t') be any law in L of positive size. If for all variables elements of $S_{D,L}$ are substituted, then t and t' yield the same value since (t, t') holds in $S_{D,L}$. Now assume that for at least one occurring variable x the value u in $S'_{D,L}$ is substituted. Then x occurs in both t and t' since (t, t') is conservative, and t and t' yield the same values as $\rho_x(t)$ and $\rho_x(t')$ respectively. These values are equal since either $\rho_x(t) = \rho_x(t')$, or $(\rho_x(t), \rho_x(t'))$ is a law in L of a smaller size which holds in $S'_{D,L}$ by the induction hypothesis. So for each substitution of variables by elements of $S'_{D,L}$ the terms t and t' yield the same value in $S'_{D,L}$. So L holds in $S'_{D,L}$.

Let $(B, i_B, \otimes)$ be any algebra containing a unit element u_B, and for which L holds. Let $\phi : S_{D,L} \to B$ be the unique homomorphism. Define $\psi : S'_{D,L} \to B$ by

$$\psi(s) = \begin{cases} \phi(s) & \text{if } s \in S_{D,L} \\ u_B & \text{if } s = u. \end{cases}$$

Then ψ is an extended homomorphism, and it is easy to see that no other extended homomorphism from $S'_{D,L}$ to $(B, i_B, \otimes)$ can exist. Hence $S'_{D,L}$ satisfies the extended universal property of $S^u_{D,L}$. □

It is possible to give a slightly less restrictive definition of unit-closedness for which the same proposition can be derived. For example, we may require

$$\{(\rho_x(t), \rho_x(t'))\} \prec_D L$$

instead of

$$\rho_x(t) = \rho_x(t') \ \text{ or } \ (\rho_x(t), \rho_x(t')) \in L.$$

Further in a simulation of replacing x by an abstract unit element one expects $(x \oplus x) \oplus (x \oplus x)$ rewrites to x, while

$$\rho((x \oplus x) \oplus (x \oplus x)) = (x \oplus x).$$

However, a definition covering these extensions will be more complicated, while our definition suffices for all purposes and for any given set of laws unit-closedness can be checked straightforward.

Let us look again to the example in which D contains only one element, and L consists of the associative law and

$$(x, \underbrace{x \oplus x \oplus \cdots \oplus x}_{n\times}).$$

This set of laws is unit-closed, so $S^u_{D,L} = S'_{D,L}$. As was already noted, $S_{D,L}$ is the cyclic group of $n-1$ elements, which contains a unit element. Let

$$\psi : S^u_{D,L} \to S_{D,L}$$

be the unique extended homomorphism, and let

$$\phi : S_{D,L} \to S^u_{D,L}$$

be the unique homomorphism. Then both $\phi \circ \psi$ and $\psi \circ \phi$ are homomorphisms, and in the style of earlier proofs one should tend to conclude that $S^u_{D,L}$ and $S_{D,L}$ are isomorphic. However, this is not true, since the one contains $n-1$ and the other n elements. The bug in this reasoning is that ϕ and $\phi \circ \psi$ are no *extended* homomorphisms: they do not map the unit element to the unit element.

We conclude this section by some more familiar examples. If L is empty, then $S_{D,L}$ corresponds to the non-empty binary trees over D. The unit element which can be added to this structure can be considered as the empty binary tree.

If L consists only of the associative law, then $S_{D,L}$ represents the non-empty finite sequences over D. The unit element which can be added to this structure can be considered as the empty finite sequence.

If L consists of both the associative and the commutative law, then $S_{D,L}$ corresponds to the non-empty finite bags over D. The unit element to be added can be considered as the empty bag.

At last, as we already mentioned, if L consists of the associative law, the commutative law and the idempotent law, then $S_{D,L}$ corresponds to the non-empty finite subsets of D. The unit element added to this structure can be considered as the empty set.

5 Lemmas and laws in Algorithmics

The universal property of $S_{D,L}$ will be used now to define some basic constructions in algorithmics like the *map, reduction* and *filter* operators, and to derive properties like the *promotion lemmas* and the *homomorphism lemma.* In practice all induction arguments can be replaced by using the definition of $S_{D,L}$. The proofs of all of these properties have the same shape: two different expressions are both homomorphisms from $S_{D,L}$ to some other algebra, hence they are equal. Loosely speaking we can say that if two expressions have the same type then they are equal.

By adding conditions on preserving the unit, all propositions and constructions in this section also hold for $S^u_{D,L}$ instead of $S_{D,L}$. Since they are completely similar, we do not mention them apart. In the next section, on the filter operator, the unit turns out to be essential.

Proposition 10 (map) *Let $f : D \to D'$ be any map and L a set of laws. Then there is exactly one map $f^* : S_{D,L} \to S_{D',L}$ such that*

- $f^* \circ i_{S_{D,L}} = i_{S_{D',L}} \circ f$, *and*
- $f^*(s \oplus s') = f^*(s) \oplus f^*(s')$ *for all* $s, s' \in S_{D,L}$.

Proof: Apply the universal property to the algebra $(S_{D',L}, i_{S_{D',L}} \circ f, \oplus)$. □

The operator $*$ which maps a function f to f^* is called the *map* operator. For example, for a function from reals to integers the map operator produces a map from the finite sequences of reals to the finite sequences of integers, which maps the finite sequences elementwise by the original function.

Proposition 11 (reduction) *Let D be a set and $\odot : D \times D \to D$. Let L be a set of laws that holds in $(D, \odot)$. Then $i_{S_{D,L}}$ is injective and there is exactly one map $\odot/ : S_{D,L} \to D$ such that*

- $\odot/ \circ i_{S_{D,L}} = id_D$, *and*
- $\odot/(s \oplus s') = (\odot/(s)) \odot (\odot/(s'))$ *for all* $s, s' \in S_{D,L}$.

Proof: Apply the universal property to $(D, id_D, \odot)$. The injectivity of $i_{S_{D,L}}$ follows from $\odot/ \circ i_{S_{D,L}} = id_D$. □

The operator / which maps an operator $\odot$ to $\odot/$ is called the *reduction* operator. As an example we mention that the familiar notations Σ and Π for sums and products over finite sequences, are nothing else than $+/$ and $\times/$ respectively.

The next proposition is the homomorphism lemma from [1,2]. Although we shall never use it, we present and prove it for the sake of completeness.

It is rather confusing that *homomorphism* has a different meaning in [1,2]. There a homomorphism is defined as an operator preserving map from $(S_{D,L}, \oplus)$ to $(D', \odot)$, where $(D', \odot)$ is arbitrary. In several branches of mathematics the word 'homomorphism' has several meanings, but always the domain and the target have the same type, and compositions of homomorphisms are homomorphisms. However, not in [1,2]. Further in [1,2] no preserving of singletons is required in the homomorphism definition, though the singletons are an essential part of the signature. Enough reasons to avoid their terminology.

Proposition 12 (homomorphism lemma) *Let D and D' be sets, $f : D \to D'$ an arbitrary map and $\odot : D' \times D' \to D'$. Let L be a set of laws that holds in $(D', \odot)$. Let $g : S_{D,L} \to D'$ be a map for which*

- $g \circ i_{S_{D,L}} = f$, *and*
- $g(s \oplus s') = g(s) \odot g(s')$ *for all* $s, s' \in S_{D,L}$.

Then $g = \odot/ \circ f^*$.

Proof: Note that

$$\odot/ \circ f^* \circ i_{S_{D,L}} \;=\; \odot/ \circ i_{S'_{D,L}} \circ f \;=\; f$$

and

$$\begin{aligned} \odot/ \circ f^*(s \oplus s') &= \odot/(f^*(s) \oplus f^*(s')) \\ &= (\odot/ \circ f^*)(s) \odot (\odot/ \circ f^*)(s') \end{aligned}$$

for all $s, s' \in S_{D,L}$. So $\odot/ \circ f^*$ is a homomorphism from $(S_{D,L}, i_{S_{D,L}}, \oplus)$ to $(D', f, \odot)$. However, the same holds for g. According to the universal property only one such a homomorphism exists, so we may conclude that

$$g = \odot/ \circ f^*.$$

□

The next three propositions are laws 1, 2, and 3 in [1]. The last two of them can be seen as corollaries of the homomorphism lemma. We prove them by only using the universal property.

Proposition 13 (composition) *Let $f : D \to D'$ and $g : D' \to D''$. Then*

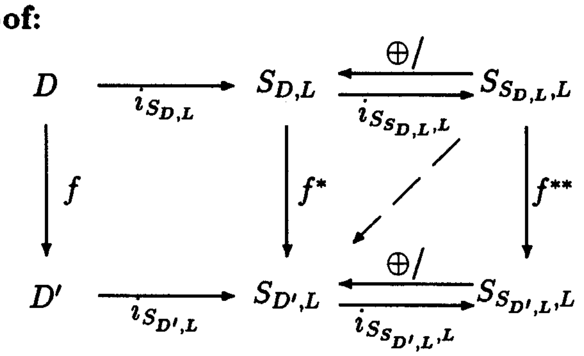
$$(g \circ f)^* = g^* \circ f^*.$$

Proof: Both expressions are homomorphisms from $S_{D,L}$ to $S_{D'',L}$, so they are equal. □

Proposition 14 (map promotion) *Let $f : D \to D'$. Then*

$$f^* \circ \oplus/ = \oplus/ \circ f^{**}.$$

Proof:



We have

$$\begin{aligned}(f^* \circ \oplus/) \circ i_{S_{S_{D,L},L}} &= f^* \\ &= \oplus/ \circ i_{S_{S_{D',L},L}} \circ f^* \\ &= (\oplus/ \circ f^{**}) \circ i_{S_{S_{D,L},L}};\end{aligned}$$

so both $f^* \circ \oplus/$ and $\oplus/ \circ f^{**}$ are homomorphisms of algebras over $S_{D,L}$ from $S_{S_{D,L},L}$ to $(S_{D',L}, f^*, \oplus)$. Hence they are equal. □

As an example let f be doubling of numbers. According to the map promotion first concatenating a sequence of sequences of numbers and then doubling their elements yields the same result as first doubling all elements and then concatenating.

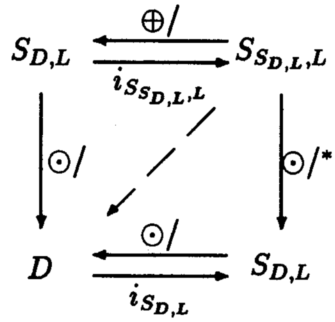
Proposition 15 (reduction promotion) *Let L be a set of laws and let*

$$\odot : D \times D \to D$$

satisfy L. Then

$$\odot/ \circ \oplus/ = \odot/ \circ \odot/^*.$$

Proof:



We have

$$\begin{aligned}(\odot/ \circ \oplus/) \circ i_{S_{S_{D,L},L}} &= \odot/ \\ &= \odot/ \circ i_{S_{D,L}} \circ \odot/ \\ &= (\odot/ \circ \odot/^*) \circ i_{S_{S_{D,L},L}};\end{aligned}$$

hence both $\odot/ \circ \oplus/$ and $\odot/ \circ \odot/^*$ are homomorphisms of algebras over $S_{D,L}$ from $S_{S_{D,L},L}$ to $(D, \odot/, \odot)$. According to the universal property they are equal. □

For example, if $\odot$ is multiplication, then the reduction promotion says that first concatenate a sequence of sequences of numbers and then multiply them yields the same result as first multiply each sequence and then multiply all of these products.

6 The filter operator

The next tool for doing algorithmics is *filter* on a predicate on D. Intuitively this is a function on $S_{D,L}$ that does not change parts for which the predicate holds and throws away the parts for which the predicate does not hold. In principle everything can be thrown away; in that case nothing will remain. In algebra terms this notion of 'nothing' corresponds to the unit element, so it is essential that the image of a filter operator contains a unit element. A natural choice for this image is $S^u_{D,L}$ as defined in section 4. We prefer to define the filter operator as a function on $S^u_{D,L}$ instead of on $S_{D,L}$. Two reasons for this are:

- Composition of filters will be possible.
- Since $S^u_{D,L}$ is an algebra over D satisfying the laws in L, there is a unique homomorphism from $S_{D,L}$ to $S^u_{D,L}$. Filter on $S_{D,L}$ can be defined by composition of this homomorphism and filter on $S^u_{D,L}$, but not conversely.

Definition 10 (filter) *Let D be a set, and let p be a predicate on D. Let L be a set of laws. Let $i_p : D \to S^u_{D,L}$ be defined as follows:*

$$i_p(d) = \begin{cases} i_{S^u_{D,L}}(d) & \text{if } p(d), \\ u & \text{if not } p(d). \end{cases}$$

Then the filter $p\triangleleft$ *is the unique extended homomorphism from*

$$(S^u_{D,L}, i_{S^u_{D,L}}, \oplus) \text{ to } (S^u_{D,L}, i_p, \oplus).$$

Before deriving some basic properties of the filter, we give an example showing that taking

$$S'_{D,L} = S_{D,L} \cup \{u\}$$

instead of $S^u_{D,L}$ would become disastrous if L is not unit-closed. Let D contain at least two elements and let L consist of

$$(x, (x \oplus y)).$$

The algebra $S_{D,L}$ does not contain a unit element; in fact it is equal to D itself. Assume there exists a filter homomorphism $p\triangleleft$ from $S'_{D,L}$ to $S'_{D,L}$, or from $S_{D,L}$ to $S'_{D,L}$. Let d, d' be two elements of D and let p be a predicate on D satisfying $\neg p(d) \wedge p(d')$. Then

$$\begin{aligned} u &= p \triangleleft i_{S'_{D,L}}(d) \\ &= p \triangleleft (i_{S'_{D,L}}(d) \oplus i_{S'_{D,L}}(d')) \\ &= p \triangleleft i_{S'_{D,L}}(d) \oplus p \triangleleft i_{S'_{D,L}}(d') \\ &= u \oplus i_{S'_{D,L}}(d') \\ &= i_{S'_{D,L}}(d') \\ &\neq u. \end{aligned}$$

Hence in this case simply adding an abstract unit element to $S_{D,L}$ does not give rise to a useful filter homomorphism.

Proposition 16 (filter commutativity) *Let p and q be predicates on D. Then*

$$p\triangleleft \circ q\triangleleft \;=\; q\triangleleft \circ p\triangleleft \;=\; (p \wedge q)\triangleleft .$$

Proof: Let $d \in D$. Then

$$\begin{aligned} q \triangleleft \circ p \triangleleft (i_{S^u_{D,L}}(d)) &= q \triangleleft (i_p(d)) \\ &= \begin{cases} q \triangleleft (i_{S^u_{D,L}}(d)) & \text{if } p(d) \\ q \triangleleft (u) & \text{if not } p(d) \end{cases} \\ &= \begin{cases} i_{S^u_{D,L}}(d) & \text{if } p(d) \text{ and } q(d) \\ u & \text{if } p(d) \text{ and not } q(d) \\ u & \text{if not } p(d) \end{cases} \\ &= i_{p\wedge q}(d). \end{aligned}$$

From the extended universal property we see that

$$p\triangleleft \circ q\triangleleft \;=\; (p \wedge q)\triangleleft .$$

Completely analogous we obtain

$$q\triangleleft \circ p\triangleleft \;=\; (p \wedge q)\triangleleft .$$

□

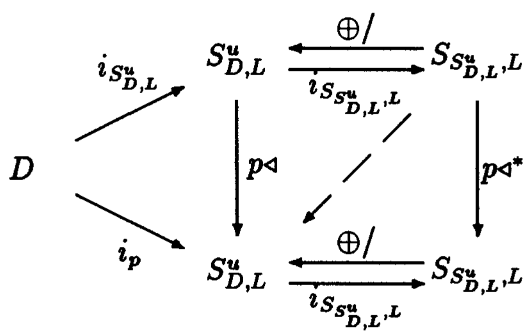
A direct consequence is **filter idempotency**: if we take $q = p$ then we get

$$p\triangleleft \circ p\triangleleft \;=\; p\triangleleft .$$

Proposition 17 (filter promotion) *Let p be be a predicate on D. Then*

$$p\triangleleft \circ \oplus/ \;=\; \oplus/ \circ p\triangleleft^{*} .$$

Proof:



We have

$$\begin{aligned}(p\triangleleft \circ \oplus/) \circ i_{S_{S^u_{D,L}},L} &= p\triangleleft \\ &= \oplus/ \circ i_{S_{S^u_{D,L}},L} \circ p\triangleleft \\ &= (\oplus/ \circ p\triangleleft^{*}) \circ i_{S_{S^u_{D,L}},L}\end{aligned}$$

so both $p\triangleleft \circ \oplus/$ and $\oplus/ \circ p\triangleleft^{*}$ are equal to the unique homomorphism from $S_{S^u_{D,L}},L$ to $(S^u_{D,L}, p\triangleleft, \oplus)$. □

Indeed, first concatenating a sequence of sequences of integers and then taking the even numbers among them yields the same result as first taking the even numbers and then concatenating the resulting sequence of sequences.

7 An example: totally ordered trees

The results until now seem rather trivial when only applied to well-known examples as trees, sequences, bags and sets. In this section we introduce a less familiar data structure to which the results can be applied: totally ordered binary trees.

Given a set of laws one can wonder how to represent the elements of the corresponding structure. One possible way is to choose for each element of the structure one particular binary tree to represent it. For a domain D, the binary trees over D are defined inductively by

$$tree ::= d \mid (tree, tree)$$

for $d \in D$.

For example, each sequence over D can be represented as a binary tree over D in which the left hand side of each subtree is an element of D. Tree composition has to be modified in order to be closed in this representation, for example the composition of $(d_1 \oplus d_2)$ and $(d_3 \oplus d_4)$ is

$$(d_1 \oplus (d_2 \oplus (d_3 \oplus d_4)))$$

instead of

$$((d_1 \oplus d_2) \oplus (d_3 \oplus d_4)).$$

A representation for bags over D can be chosen depending on some total order on D. Then a bag over D can be seen as a non-decreasing sequence of elements of D, while composition corresponds to merging of such sequences. At last, a finite subset of D can be represented by a strictly increasing sequence, while composition corresponds to merging and removing double elements.

In a similar way we shall introduce totally ordered trees; then we shall show that the algebra of totally ordered trees over D is exactly $S^u_{D,L}$ for

$$L = \{((x_1 \oplus x_2), (x_2 \oplus x_1)), ((x_1 \oplus x_1), (x_2 \oplus x_2))\}.$$

Note that the second law in L is not conservative and that L is not unit-closed. In fact this is an example for $S'_{D,L}$ and $S^u_{D,L}$ being not isomorphic, as we show now. Let x be a non unit element in $S'_{D,L}$, then $x \in S_{D,L}$, so also $x \oplus x \in S_{D,L}$, so

$$x \oplus x \neq u.$$

However, for each element x in $S^u_{D,L}$ we have

$$x \oplus x = u \oplus u = u,$$

so $S'_{D,L}$ and $S^u_{D,L}$ are not isomorphic.

As for the bag representation, assume a total order $<$ on D is given. This total order can be extended inductively to a total order on binary trees over D:

$$\begin{array}{ccc} d_1 < (t_1, t_2) & \equiv & \textbf{true} \\ (t_1, t_2) < d_1 & \equiv & \textbf{false} \\ (t_1, t_2) < (t_3, t_4) & \equiv & t_1 < t_3 \vee (t_1 = t_3 \wedge t_2 < t_4) \end{array}$$

for all $d_1, d_2 \in D$ and for all binary trees t_1, t_2, t_3, t_4 over D.

The *totally ordered trees* over D are defined as a subset of the binary trees over D inductively as follows:

- elements of D are totally ordered trees, and
- if t_1, t_2 are totally ordered trees with $t_1 < t_2$, then (t_1, t_2) is a totally ordered tree.

If not $t_1 < t_2$, then (t_1, t_2) is not a totally ordered tree. So these totally ordered trees are not closed under common tree composition. However, by adding an abstract unit element and adjusting tree composition the algebra of totally ordered trees can be constructed in a natural way, as is done in the next definition.

Definition 11 *For a totally ordered set D the algebra $\mathcal{T} = (TOT, i_{TOT}, \oplus)$ over D is defined by:*

- *TOT consists of the totally ordered trees over D and an abstract element u;*
- *$i_{TOT}(d) = d$ for all $d \in D$;*
- $$\begin{aligned} u \oplus u &= u, \\ u \oplus t_1 &= t_1, \\ t_1 \oplus u &= t_1, \\ t_1 \oplus t_2 &= \begin{cases} u & \text{if } t_1 = t_2 \\ (t_1, t_2) & \text{if } t_1 < t_2 \\ (t_2, t_1) & \text{if } t_2 < t_1 \end{cases} \end{aligned}$$

  for all totally ordered trees t_1, t_2.

Proposition 18 *Let D be a totally ordered set and let $\mathcal{T}$ be as above. Let*

$$L = \{((x_1 \oplus x_2), (x_2 \oplus x_1)), ((x_1 \oplus x_1), (x_2 \oplus x_2))\}.$$

Then $\mathcal{T}$ is equal to $S^{u}_{D,L}$, up to extended isomorphism.

Proof: Clearly TOT satisfies L. For an arbitrary algebra

$$\mathcal{A} = (A, i_A, \odot)$$

over D containing a unit element u_A, we have to prove that there is exactly one extended homomorphism

$$\phi : \mathcal{T} \to \mathcal{A}.$$

Each element of TOT can be written either as u or as a binary tree in a unique way, so we can define ϕ as a map from TOT to A as follows:

$$\begin{aligned} \phi(u) &= u_A \\ \phi(d) &= i_A(d) \\ \phi((t_1, t_2)) &= \phi(t_1) \odot \phi(t_2) \end{aligned}$$

for each $d \in D$ and for all totally ordered trees t_1, t_2 with $t_1 < t_2$. Then

$$\phi(u \oplus u) = \phi(u) = u_A = u_A \odot u_A = \phi(u) \odot \phi(u),$$

$$\phi(u \oplus t_1) = \phi(t_1) = u_A \odot \phi(t_1) = \phi(u) \odot \phi(t_1),$$

$$\phi(t_1 \oplus u) = \phi(t_1) = \phi(t_1) \odot u_A = \phi(t_1) \odot \phi(u),$$

$$\phi(t_1 \oplus t_2) = \begin{cases} \phi(u) = u_A = u_A \odot u_A = \phi(t_1) \odot \phi(t_1) = \phi(t_1) \odot \phi(t_2) & \text{if } t_1 = t_2 \\ \phi((t_1, t_2)) = \phi(t_1) \odot \phi(t_2) & \text{if } t_1 < t_2 \\ \phi((t_2, t_1)) = \phi(t_2) \odot \phi(t_1) = \phi(t_1) \odot \phi(t_2) & \text{if } t_2 < t_1 \end{cases},$$

for all totally ordered trees t_1, t_2, using that $\mathcal{A}$ satisfies L. We conclude that

$$\phi(t_1 \oplus t_2) = \phi(t_1) \odot \phi(t_2)$$

for all $t_1, t_2 \in TOT$, so ϕ is an extended homomorphism. Conversely, let $\psi : \mathcal{T} \to \mathcal{A}$ be an arbitrary extended homomorphism. Then

$$\psi(u) = u_A,$$

$$\psi(d) = i_A(d),$$

$$\psi((t_1, t_2)) = \psi(t_1 \oplus t_2) = \psi(t_1) \odot \psi(t_2)$$

for each $d \in D$ and for all totally ordered trees t_1, t_2 with $t_1 < t_2$. So ψ satisfies the definition of ϕ, so $\psi = \phi$, and ϕ is the only extended homomorphism from $\mathcal{T}$ to $\mathcal{A}$, which we had to prove. □

As a result of this proposition, the constructions map, reduction and filter are meaningful on totally ordered trees. For example, if D consists of the non-negative integers and $\otimes$ is defined by

$$d_1 \otimes d_2 \;=\; |\, d_1 - d_2 \,|,$$

then $(D, \otimes)$ satisfies L and contains a unit element, so

$$\otimes / : TOT \to D$$

is meaningful and satisfies our properties like reduction promotion.

It is quite challenging to understand this reduction promotion, map promotion and filter promotion on totally ordered trees of totally ordered trees; they seem to be far less evident than on sequences of sequences!

8 The construction

Until now we have presented various properties of $S_{D,L}$, provided that it exists, and we have given some examples for particular L. In this section the existence will be shown for arbitrary L: we show how for general D and L an algebra can be constructed satisfying the $S_{D,L}$-property.

The most simple case we get when there are no laws at all, i. e. L is the empty set. Then we can take the set of all binary trees with leaves in D. In other words if D is given, then we define $S_{D,\emptyset}$ by

$$\begin{array}{rcl} d & : & D, \\ s & : & S_{D,\emptyset}, \\ s & ::= & d \mid (s \oplus s). \end{array}$$

The map $i_{S_{D,\emptyset}} : D \to S_{D,\emptyset}$ is the natural embedding of D.

We now have to check that the universal property holds for this particular $S_{D,\emptyset}$. Let $(T, i_T, \otimes)$ be an arbitrary algebra, i. e. $i_T : D \to T$ and $\otimes : T \times T \to T$. Then by definition a homomorphism $\phi : S_{D,\emptyset} \to T$ has to satisfy

- $\phi(d) = i_T(d)$ for all $d \in D$, and
- $\phi(s \oplus s') = \phi(s) \otimes \phi(s')$ for all $s, s' \in S_{D,\emptyset}$.

The existence and uniqueness of such a ϕ follows from the definition of $S_{D,\emptyset}$, so the universal property holds.

The next thing to do is to construct $S_{D,L}$ from $S_{D,\emptyset}$ for non-empty set of laws L. For shorthand, let us write S instead of $S_{D,\emptyset}$. By definition, an element of L is a pair of terms on variables $x_1, x_2, \cdots, x_n$. If the variables $x_1, x_2, \cdots, x_n$ are replaced by elements of S and the abstract operator of terms is replaced by the operator $\oplus$ of S, then a pair of elements of S is obtained. Let L_S denote the subset of $S \times S$ consisting of all pairs of elements of S that can be obtained in this way from elements of L. Let $\bar{L}_S$ be the reflexive symmetric transitive substitutive closure of the relation L_S, i. e. $\bar{L}_S$ is the smallest subset of $S \times S$ such that

1. $L_S \subset \bar{L}_S$,
2. $(s, s) \in \bar{L}_S$ for all $s \in S$,
3. if $(s, s') \in \bar{L}_S$, then $(s', s) \in \bar{L}_S$,
4. if $(s, s'), (s', s'') \in \bar{L}_S$, then $(s, s'') \in \bar{L}_S$,
5. if $(s, s'), (\tilde{s}, \tilde{s}') \in \bar{L}_S$, then $(s \oplus \tilde{s}, s' \oplus \tilde{s}') \in \bar{L}_S$.

In other words, $\bar{L}_S$ consists of all elements of $S \times S$ that can be obtained from elements of

$$L_S \cup \{(s, s) \mid s \in S\}$$

by applying properties (3) to (5) a finite number of times.

Properties (2) to (4) imply that $\bar{L}_S$ is an equivalence relation on S. Let $\bar{S}$ denote S modulo this equivalence relation, i. e. $\bar{S}$ is the set of the corresponding equivalence classes. This set $\bar{S}$ will become an algebra satisfying the $S_{D,L}$-property. Before we can consider it as an algebra, first a map $i_{\bar{S}}$ from D to $\bar{S}$ and a binary operation on $\bar{S}$ have to be defined. For each $d \in D$ we define

$$i_{\bar{S}}(d) \;=\; \text{the equivalence class of } i_S(d).$$

Property (5) implies that the equivalence relation $\bar{L}_S$ is also a congruence relation, i. e. $\oplus$ induces a binary operation on $\bar{S}$. Choosing this binary operation we can speak about the algebra $\bar{S}$ now. Property (1) implies that the algebra obtained in this way satisfies L.

In a straightforward way it can be checked that for an arbitrary algebra $\mathcal{A}$ satisfying L there is exactly one homomorphism from $\bar{S}$ to $\mathcal{A}$; for details we refer to [4], theorem 6, where this construction is given in a more general context with a complete proof. Hence the algebra $\bar{S}$ satisfies the $S_{D,L}$-property, and the existence of $S_{D,L}$ has been proved.

Starting with

$$s \;::=\; u \mid d \mid (s \oplus s)$$

instead of

$$s \;::=\; d \mid (s \oplus s)$$

and adding the laws

$$((x \oplus u), x) \;\text{ and }\; ((u \oplus x), x)$$

to L yields the construction of $S^u_{D,L}$ in a similar way.

References

[1] L.G.L.T. Meertens. *Algorithmics, towards programming as a mathematical activity.* Mathematics an Computer Science, Proceeding CWI symp. Nov 1983. CWI Monographs Vol 1. North Holland, 1986.

[2] R. S. Bird. *An introduction to the theory of lists.* Technical Monograph PRG-56, Oxford, 1986.

[3] J. Meseguer, J. A. Goguen. *Initiality, induction and computability.* Algebraic methods in semantics, Cambridge University Press, 1985.

[4] J. A. Goguen, J. W. Thatcher, E. G. Wagner. *An initial algebra approach to the specification, correctness and implementation of abstract data types.* Current trends in programming methodology, vol 4: Data structuring. Prentice Hall, 1978.