Theoretical Computer Science 70 (1990) 3-34
North-Holland

ALGEBRAS, POLYNOMIALS AND PROGRAMS

Eric G. WAGNER

Computer Science Principles, Mathematical Sciences Department, IBM Research Division, T.J. Watson Research Center, Yorktown Heights, NY 10598, USA

Abstract. This paper presents an introduction to some of the more algebraic applications of elementary category theory in computer science. Topics include: a category based look at universal algebra; the definition of polynomials over arbitrary algebras and their application to the study of substitution; a development of Lawvere algebraic theories based on polynomials, and the application of such theories to algebra and to the study of iteration and recursion in programming languages.

Contents

0. Introduction

This paper is directed toward computer scientists who know something of the basics of universal algebra and category theory and who are interested in seeing something of the interaction of the two subjects in computer science at a level beyond the well-known applications of initiality to semantics and data type specification. The paper assumes knowledge of definitions for the concepts of category, functor, natural transformation, and adjunction.

The first section provides a very brief introduction to the concept of a Σ-algebra, a Σ-homomorphism, and an equational presentation. We give the conventional definitions but we also restate them in a more categorical manner, and we exploit a bit of category theory to nail down the notion of an algebra satisfying a set of equations.

Section 2 goes considerably further into the subject. As an introduction to universal algebra it may seem somewhat unusual, but the idea is to show that after a relatively

0304-3975/90/$3.50

small amount of set-theoretic scene setting, most of the definitions and arguments can be done in terms of categorical concepts. The approach is based on many discussions that Jesse Wright and I had some years back while we were going over some of the work of Lawvere, and of Herrlich and Strecker [9]. However, none of the aforementioned should be held responsible for what happens here. The choice of topics in the section does not provide a balanced treatment of the universal algebra, for that one should go to Cohn [5], Grätzer [8], Herrlich and Strecker [9], or McKenzie et al. [11]. But I have provided everything needed for the rest of the paper.

The third section is concerned with polynomials over arbitrary algebras. This is not a new topic to mathematicians, and, indeed, is just the natural generalization of the polynomials (over rings) that one probably first encountered in high school algebra. What is important about polynomials from a computer science viewpoint is that they provide a natural approach to the concept of variables, and the associated concept of substitution. We show, by means of very simple categorical proofs, that substitution is associative and has identities. These proofs are excellent examples of how categorical methods can cut right to the heart of the matter. It is also worth pointing out that the approach taken here need not be restricted to polynomials over algebras, and there are computer science applications for polynomials over other structures (e.g. over sketches [1]).

In the fourth section we give a treatment of Lawvere algebraic theories using categories of polynomials as a starting point. We give the definition of an algebra as a functor from an algebraic theory into the category of sets, and relate this to our earlier definition of an algebra. Algebraic theories have been used quite extensively in theoretical computer science by researches studying iteration and recursion and by a number of groups studying data type specification. We feel that polynomials provide a good approach for computer scientists to algebraic theories.

In the final section we employ polynomials and algebraic theories to model simple languages of flowcharts and monadic recursion schemes. Further examples and additional references will be found in [14].

A few points of notation: We write ω for the set $\omega = \{0, 1, \ldots\}$ of natural numbers. For any $n \in \omega$ we write $[n]$ for the set $[n] = \{1, \ldots, n\}$, so, in particular, $[0] = \emptyset$. We define a string α of length n on alphabet A as a mapping $\alpha : [n] \to A$. We write ε to denote the empty string $\varepsilon : [0] \to A$. We also write $F \dashv G : A \to B$ as a shorthand for the functor $F : B \to A$ is a left adjoint for the functor $G : A \to B$.

1. Universal algebra: basic concepts

1.1. Definition. Given a set S define $\mathbf{Set}^S$, *the category of S-sorted sets*, to have, as objects, all functors from the discrete category S to **Set**, and to have, as morphisms, all the associated natural transformations. To rephrase this in a less categorical fashion, the S-sorted sets are S-indexed families, $\Omega = \langle \Omega_s \mid s \in S \rangle$, of sets, and for

S-sorted sets Ω_1 and Ω_2, a morphism h from Ω_1 to Ω_2 is an S-indexed family of mappings, $h = \langle h_s : (\Omega_1)_s \to (\Omega_2)_s \mid s \in S \rangle$.

1.2. Definition. Let S be a set (of *sorts*), then by an *S-sorted signature* we mean an $(S^* \times S)$-sorted set $\Sigma \in \mathbf{Set}^{S^* \times S}$, that is, an $(S^* \times S)$-indexed family, $\Sigma = \langle \Sigma_{w,s} \mid w \in S^*, s \in S \rangle$, of sets. An element $\sigma \in \Sigma_{w,s}$ is said to be an *operator symbol* of *rank* $\langle w, s \rangle$, *arity* w, and sort s. An element of $\Sigma_{\varepsilon,s}$ is sometimes called a *constant operator of sort* s. It is convenient to regard the sets $\Sigma_{w,s}$ as being pairwise disjoint.

1.3. Definition. Let Σ be an S-sorted signature. Then a *Σ-algebra A* consists of an S-sorted set $|A| = \langle A_s \mid s \in S \rangle \in |\mathbf{Set}^S|$, together with, for each $w = s_1 \ldots s_n \in S^*$, $s \in S$, and $\sigma \in \Sigma_{w,s}$, a function

$$\sigma_A : A_{s_1} \times A_{s_2} \times \cdots \times A_{s_n} \to A_s.$$

The set A_s is called the *carrier of sort* s, and the function σ_A is called the *operation of A named by* σ. For $\sigma \in \Sigma_{\varepsilon,s}$ we have $\sigma_A \in A_s$ (also written $\sigma_A : \to A_s$). We call $\{\sigma_A \mid \sigma \in \Sigma_{\varepsilon,s}\}$ the *set of named constants* of A of sort s.

Generally, for $w = s_1 s_2 \ldots s_n \in S^*$, we shall write A^w to denote $A_{s_1} \times A_{s_2} \times \cdots \times A_{s_n}$.

Let A and B be Σ-algebras, then a *Σ-homomorphism* $h : A \to B$ is a family of functions $h = \langle h_s : A_s \to B_s \mid s \in S \rangle$ such that:

If $\sigma \in \Sigma_{\varepsilon,s}$ then $h_s(\sigma_A) = \sigma_B$.

If $\sigma \in \Sigma_{w,s}$, with $w = s_1 s_2 \ldots s_n$ and $\langle a_1, \ldots, a_n \rangle \in A^w$, then

$$h_s(\sigma_A(a_1, \ldots, a_n)) = \sigma_B(h_{s_1}(a_1), \ldots, h_{s_n}(a_n)).$$

Given a signature Σ, let $\mathbf{Alg}_\Sigma$ denote the category whose class of objects is the class of all Σ-algebras and whose morphisms are the homomorphisms between the Σ-algebras.

The above definition can be stated purely in terms of many-sorted sets and many-sorted mappings. First some notation. Given a set S, let $\theta : \mathbf{Set}^S \to \mathbf{Set}^{S^*}$ such that for $\Omega : S \to \mathbf{Set}$, $\theta(\Omega) = \Omega^*$ where, if $w = s_1 \ldots s_n$, then $\Omega^*(w) = \Omega(s_1) \times \cdots \times \Omega(s_n)$, and, if $h : \Omega_1 \to \Omega_2$, $h = \langle h_s : \Omega_1(s) \to \Omega_2(s) \mid s \in S \rangle$, then $\theta(h) = h^* = \langle (h^*)_w : \Omega_1^*(w) \to \Omega_2^*(w) \mid w \in S^* \rangle$, $(h^*)_w = h_{s_1} \times \cdots \times h_{s_n}$. Secondly, for each S-sorted Ω, let

$$\mathbf{Sem}_\Omega : (S^* \times S) \to \mathbf{Set}, \qquad \langle w, s \rangle \mapsto \mathbf{Set}(\Omega^*(w), \Omega(s)).$$

Then, for any S-sorted signature $\Sigma \in \mathbf{Set}^{S^* \times S}$, a Σ-algebra consists of an S-sorted set $|A| : S \to \mathbf{Set}$ together with an $(S^* \times S)$-sorted mapping $A : \Sigma \to \mathbf{Sem}_{|A|}$, and a homomorphism from $\langle |A|, A \rangle$ to $\langle |B|, B \rangle$ is an S-sorted mapping $h : |A| \to |B|$ such that the diagram

$$\begin{tikzcd} |A|(w) \arrow[r, "A(\sigma)"] \arrow[d, "(h^*)_w"'] & |A|(s) \arrow[d, "(h^*)_s = h_s"] \\ |B|(w) \arrow[r, "B(\sigma)"'] & |B|(s) \end{tikzcd}$$

commutes for all $\sigma \in \Sigma_{w,s}$.

1.4. Definition. An object I in a category $\boldsymbol{C}$ is said to be *initial* if for every object A in $\boldsymbol{C}$ there is exactly one morphism $i_A : I \to A$.

1.5. Proposition. *For any signature Σ the category, $\mathbf{Alg}_\Sigma$ of all Σ-algebras has initial objects. Indeed, given an S-sorted signature Σ, then one of the initial objects is the Σ-algebra $\mathcal{I}nit_\Sigma$ such that, for each $s \in S$, $(\mathcal{I}nit_\Sigma)_s$ is the smallest set of strings on the alphabet $\bar{\Sigma} = \bigcup \langle \Sigma_{w,s} \mid w \in S^*, s \in S \rangle \cup \{(,)\}$ such that:*

(i) *If $\sigma \in \Sigma_{\varepsilon,s}$, then the string $\sigma \in (\mathcal{I}nit_\Sigma)_s$. Where, as a string, σ is the mapping $\sigma : [1] \to \bar{\Sigma}$, $1 \mapsto \sigma$.*

(ii) *If $\sigma \in \Sigma_{w,s}$, where $w = s_1 \ldots s_n$ and $t_i \in (\mathcal{I}nit_\Sigma)_{s_i}$, for $i = 1, \ldots, n$, then the string $\sigma(t_1 t_2 \ldots t_n) \in (\mathcal{I}nit_\Sigma)_s$.*

Finally, for any $\sigma \in \Sigma_{w,s}$, with $w = s_1 \ldots s_n$, define

$$\sigma_{\mathcal{I}nit} : (\mathcal{I}nit_\Sigma)_{s_1} \times \cdots \times (\mathcal{I}nit_\Sigma)_{s_n} \to (\mathcal{I}nit_\Sigma)_s, \qquad \langle t_1, \ldots, t_n \rangle \mapsto \sigma(t_1 \ldots t_n)$$

for any $t_i \in (\mathcal{I}nit_\Sigma)_{s_i}$, $i = 1, \ldots, n$. $\mathcal{I}nit_\Sigma$ is called the word algebra *on Σ.*

1.6. Definition. Let Σ be an S-sorted signature and let $X = \langle X_s \mid s \in S \rangle$ be an S-indexed family of sets (of *variables*) disjoint from all the $\Sigma_{s,w}$. Define $\Sigma(X)$ to be the signature with $\Sigma(X)_{\varepsilon,s} = \Sigma_{\varepsilon,s} \cup X_s$ and, for $w \neq \varepsilon$, $\Sigma(X)_{w,s} = \Sigma_{w,s}$. Let $\mathcal{I}nit_{\Sigma(X)}$ be the word algebra on $\Sigma(X)$, then we can also view $\mathcal{I}nit_{\Sigma(X)}$ as a Σ-algebra, which we will write as $\mathcal{F}ree_\Sigma(X)$, and call *the Σ-algebra freely generated by X*. More precisely, $\mathcal{F}ree_\Sigma(X)$ is the Σ-algebra, where the carrier $(\mathcal{F}ree_\Sigma(X))_s$ of sort s is the set of strings $(\mathcal{I}nit_{\Sigma(X)})_s$, and for $\sigma \in \Sigma_{w,s}$, $w = s_1 \ldots s_n$, $\sigma_{\mathcal{F}ree_\Sigma}(x) = \sigma_{\mathcal{I}nit_{\Sigma(X)}}$; the difference between the two algebras is that the elements of X are constant operators in $\mathcal{I}nit_{\Sigma(X)}$, but only elements of the carrier in $\mathcal{F}ree_\Sigma(X)$.

The important property of such a free algebra is given by the following proposition.

1.7. Proposition. *Let Σ and X be as above and let A be any Σ-algebra, then any family of mappings $h = \langle h_s : X_s \to A_s \mid s \in S \rangle$ extends uniquely to a homomorphism $h^\# : \mathcal{F}ree_\Sigma(X) \to A$. Put another way, let U_Σ, the* underlying set functor, *be the functor*

$$U_\Sigma : \mathbf{Alg}_\Sigma \to \mathbf{Set}^S, \qquad A \mapsto \langle A_s \mid s \in S \rangle, \qquad h \mapsto \langle h_s \mid s \in S \rangle,$$

then U_Σ has a left adjoint $F_\Sigma : \mathbf{Set}^S \to \mathbf{Alg}_\Sigma$ which takes $X \in |\mathbf{Set}^S|$ to $F_\Sigma(X) = \mathscr{F}\!\mathit{ree}_\Sigma(X)$. That is, there exists a S-ary mapping $\eta_X : X \to U_\Sigma(\mathscr{F}\!\mathit{ree}_\Sigma(X))$, such that for each Σ-algebra A, and each S-ary mapping $h : X \to U_\Sigma(A)$, there is a unique homomorphism $h^\# : \mathscr{F}\!\mathit{ree}_\Sigma(X) \to A$ such that $U_\Sigma(h^\#) \bullet \eta_X = h$.

$$\begin{tikzcd} X \arrow[r, "\eta_X"] \arrow[dr, "h"'] & U_\Sigma(\mathscr{F}\!\mathit{ree}_\Sigma(X)) \arrow[d, "U(h^\#)"] & \mathscr{F}\!\mathit{ree}_\Sigma(X) \arrow[d, "h^\#"] \\ & U_\Sigma(A) & A \end{tikzcd}$$

Proof. $U_\Sigma(\mathscr{F}\!\mathit{ree}_\Sigma(X))$ is, of course, the S-ary set of strings $\langle (\mathscr{I}\!\mathit{nit}_{\Sigma(X)})_s \mid s \in S \rangle$. Let $\eta_X : X \to U_\Sigma(\mathscr{F}\!\mathit{ree}_\Sigma(X))$ be the S-ary mapping each $x \in X$ to the string x. Then $h^\#$ is completely determined since it must take $\sigma \in \Sigma_{w,s}$ to σ_A, and it must take $x \in X$ to $h(x)$. The detailed proof proceeds by induction on the definition of $\mathscr{I}\!\mathit{nit}_{\Sigma(X)}$. □

We next introduce the concepts of equationally defined classes of Σ-algebras, that is, loosely speaking, classes of algebras satisfying given sets of equations. We apply the above machinery to give a clean definition of equations and their satisfaction.

1.8. Definition. A *Σ-equation* is a pair $e = \langle L, R \rangle$ where $L, R \in (\mathscr{F}\!\mathit{ree}_\Sigma(X))_s$ for some s. Let $\mathrm{var}(e) = \mathrm{var}(L) \cup \mathrm{var}(R)$ (union as S-indexed families) be the family of variables occurring in e. Let A be a Σ-algebra and let e be a Σ-equation, then we say A *satisfies* e iff $\theta^\#(L) = \theta^\#(R)$ for all *assignments* $\theta : \mathrm{var}(e) \to U_\Sigma(A)$. If E is a set of Σ-equations, then A *satisfies* E iff A satisfies every $e \in E$.

1.9. Definition. By an *equational presentation* (or just a *presentation*) we mean a triple $P = \langle S, \Sigma, E \rangle$ where Σ is an S-sorted signature and E is a set of Σ-equations. Let $P = \langle S, \Sigma, E \rangle$ be a presentation, then a Σ-algebra satisfying E is called a P-algebra, and the subcategory of $\mathbf{Alg}_\Sigma$ consisting of all the P-algebras together with all the homomorphisms between them, is denoted $\mathbf{Alg}_P$. When S is evident from context, we may write a presentation as a pair $\langle \Sigma, E \rangle$.

1.10. Example. To help with the motivation for Section 3 we present a "classical" example of an equationally defined category of algebras, namely the category of *rings* (or, more precisely, the category of commutative rings with units). A ring is a one-sorted algebra (i.e. S is a singleton set, $S = \{s\}$) with presentation $\mathbf{Ring} = \langle S, \Sigma, E \rangle$ where Σ is the signature with

$$\Sigma_{\lambda,s} = \{0, 1\}, \qquad \sigma_{s,s} = \{-\}, \qquad \Sigma_{ss,s} = \{+, *\}$$

and E is the set of equations (written taking $X=\{a, b, c\}$):

$$(a+b)+c=a+(b+c),$$

$$a+0=a, \qquad a+(-a)=0, \qquad a+b=b+a,$$

$$(a*b)*c=a*(b*c), \qquad a*1=a, \qquad a*b=b*a,$$

$$a*(b+c)=(a*b)+(a*c),$$

$$(a+b)*c=(a*c)+(b*c).$$

1.11. Example. We give a signature for the syntax of the primitive operations of a toy programming language.

$$S=\{\mathrm{INT}, \mathrm{BOOL}\},$$

$$\Sigma_{\lambda,\mathrm{INT}}=\{0, 1, 2, \ldots\},$$

$$\Sigma_{\lambda,\mathrm{BOOL}}=\{\mathit{true}, \mathit{false}\},$$

$$\Sigma_{\mathrm{INT},\mathrm{INT}}=\{S, P, -\},$$

$$\Sigma_{\mathrm{INT},\mathrm{INT},\mathrm{INT}}=\{+, *, -\},$$

$$\Sigma_{\mathrm{INT},\mathrm{INT},\mathrm{BOOL}}=\{\mathrm{Eq}\},$$

$$\Sigma_{\mathrm{BOOL},\mathrm{BOOL}}=\{\neg\},$$

$$\Sigma_{\mathrm{BOOL},\mathrm{BOOL},\mathrm{BOOL}}=\{\wedge, \vee\},$$

$$\Sigma_{\mathrm{BOOL},\mathrm{INT},\mathrm{INT},\mathrm{INT}}=\{\mathrm{If}\}.$$

Such a signature can be presented pictorially as shown in Fig. 1. Such a picture is sometimes called a *spaghetti and meatball diagram.* Let **Prim** denote the presentation **Prim** $=\langle S, \Sigma, \emptyset\rangle$ with the above signature and no axioms. Let **FPr** denote the initial **Prim**-algebra $\mathcal{I}nit_{\mathbf{Prim}}$, and let **IPr** denote the **Prim**-algebra corresponding to the "usual interpretation" of the operators and sorts in **Prim**. Finally let In: **FPr** → **IPr** denote the unique homomorphism from **FPr** to **IPr** given by the initiality of **FPr**.

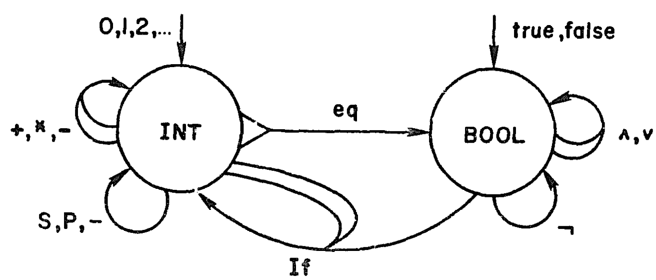

Fig. 1.

Declaring **IPr** to be the "usual interpretation" of **FPr**, is somewhat vague, and it is meant to be so. The fact is, in computer science, the INTegers, that is, the interpretation of INT, have a very wide range of interpretations. This is, of course, a result of the fact that the "INTeger type" built into a computer is some finite approximation of the mathematical integers, and most (though not all) programming languages employ the version of the INTegers, and the arithmetic operations, built into the computer on which they are implemented. By not axiomatizing the arithmetic operations we provide a treatment that will work with any of these versions.

2. More about algebras: from a categorical viewpoint

As noted in the introduction, the aim of this chapter is to provide an introduction to universal algebra which also illustrates the use of simple categorical arguments. The proofs are somewhat sketchy, especially so when the arguments are largely set theoretic.

2.1. Proposition. *A homomorphism $h: A \to B$ in* $\mathbf{Alg}_\Sigma$ *is a monomorphism iff it is injective. That is, $m: A \to B$ is a monomorphism iff for each $s \in S$, the mapping $m_s: A_s \to B_s$ is injective.*

Proof. Clearly an injective homomorphism is a monomorphism. So let $m: A \to B$ be a monomorphism in $\mathbf{Alg}_\Sigma$. Assume m is not injective. Then there must exist $s \in S$, and $a_1, a_2 \in A_s$ such that $a_1 \neq a_2$ but $m_s(a_1) = m_s(a_2)$. Let X be a S-ary set where X_s contains a single element x, and $X_t = \emptyset$ for $t \neq s$. Let

$$h1, h_2: \mathcal{T}ree_\Sigma(X) \to A, \qquad h_i: x \mapsto a_i.$$

Now, $m(h_1(x)) = m(a_1) = m(a_2) = m(h_2(x))$, so $m \bullet h_1 = m \bullet h_2$ since, by Proposition 1.7, they are completely determined by the image of x. But then, since m is a monomorphism, $h_1 = h_2$, a contradiction. □

The above result holds as well for monomorphism between P-algebras, but to prove this we need the counterpart of Proposition 1.7 for P-algebras.

2.2. Definition. Given Σ-algebras A and B define *the product of A and B, $A \times B$,* to be the Σ-algebra with carrier

$$U_\Sigma(A \times B) = U_\Sigma(A) \times U_\Sigma(B)$$

(the Cartesian product of the sets A and B), and such that, for any $w = s_1 \ldots s_n \in S^*$, $\sigma \in \Sigma_{w,s}$, and $\langle\langle a_1, b_1\rangle, \ldots, \langle a_n, b_n\rangle\rangle \in U_\Sigma(A \times B)^w$,

$$\sigma_{A \times B}(\langle\langle a_1, b_1\rangle, \ldots, \langle a_n, b_n\rangle\rangle) = \langle \sigma_A(a_1, \ldots, a_n), \sigma_B(b_1, \ldots, b_n)\rangle.$$

Define the product projections to be

$$p_{A,B}: A \times B \to A, \qquad \langle a, b \rangle \mapsto a$$

and

$$q_{A,B}: A \times B \to B, \qquad \langle a, b \rangle \to b.$$

2.3. Fact. *$\langle p_{A,B}: A \times B \to A, q_{A,B}: A \times B \to B \rangle$ is a categorical product in* $\mathbf{Alg}_\Sigma$.

2.4. Definition. Let A and B be Σ-algebras: then we say that A is a *subalgebra* of B, if $U_\Sigma(A) \subseteq U_\Sigma(B)$ and, for every $\sigma \in \Sigma_{w,s}$, and all $\bar{a} \in A^w$, $\sigma_A(\bar{a}) = \sigma_B(\bar{a})$. It is easy to see then that the S-ary inclusion mapping given by the family $\iota = \langle \iota_s : A_s \to B_s \mid s \in S \rangle$ of inclusion mappings is a morphism, indeed a monomorphism, in $\mathbf{Alg}_\Sigma$. This might suggest that the appropriate categorical generalization of subalgebra is monomorphism. This does not work since it produces too many "subalgebras". The appropriate generalization is to view a subalgebra as an isomorphism class of subalgebras. This is, in fact, the appropriate categorical generalization for subobjects in any category (see [12]).

2.5. Definition. Let A be a Σ-algebra, then by a *congruence on A* we mean a subalgebra $Q \subseteq A \times A$ such that $U_\Sigma(Q)$ is an equivalence relation on A.

2.6. Proposition. *Let $h: A \to B$ in* $\mathbf{Alg}_\Sigma$, *and let $\langle p_{A,A}: A \times A \to A, q_{A,A}: A \times A \to A \rangle$ be the indicated product. Then $h \bullet p_{A,A}$ and $h \bullet q_{AA}$ have an equalizer $e_h: Q_h \to A \times A$, which is a congruence on A.*

$$\begin{tikzcd} Q_h \arrow[r, "e_h"] & A \times A \arrow[r, bend left, "p_{A,A}"] \arrow[r, bend right, "q_{A,A}"'] & A \arrow[r, "h"] & B \end{tikzcd}$$

Proof. Take Q_h to have, as carrier $U_\Sigma(Q_h)$, the S-ary set of all pairs $\langle a_1, a_2 \rangle \in U_\Sigma(A \times A)$ such that $h(a_1) = h(a_2)$. Because $p_{A,A}$ and $q_{A,A}$ are projections, this is precisely the set of pairs $\langle a_1, a_2 \rangle$ such that

$$h \bullet p_{A,A}(a_1, a_2) = h \bullet q_{A,A}(a_1, a_2).$$

Let e_h be the inclusion mapping of $U_\Sigma(Q_h)$ into $U_\Sigma(A \times A)$. $\square$

The congruence $e_h: Q_h \to A \times A$ defined in Proposition 2.6 for $h: A \to B$ is called *the congruence induced by h.*

2.7. Proposition. *Let $e: Q \to A \times A$ be a congruence on A, then there exists a Σ-algebra denoted A/Q, and a surjective homomorphism $k_Q: A \to A/Q$, such that k_Q is a coequalizer for $p_{A,A} \bullet e$ and $q_{A,A} \bullet e$.*

$$\begin{tikzcd} Q \arrow[r, "e"] & A \times A \arrow[r, bend left, "p_{A,A}"] \arrow[r, bend right, "q_{A,A}"'] & A \arrow[r, "k_Q"] & A/Q \end{tikzcd}$$

Proof. Let the carrier, $U_\Sigma(A/Q)$ consist of the set of $U_\Sigma(Q)$-equivalence classes of $U_\Sigma(A)$. Given $\sigma \in \Sigma_{w,s}$ and $\bar{a} \in (A/Q)^w$, define $\sigma_{A/Q}(\bar{a})$ to be the $U_\Sigma(Q)$-equivalence class of $\sigma_A(a)$ where a is any element of $\bar{a}$. That this is well defined follows from Q being a subalgebra of $A \times A$. Let k_Q be given by the mapping that takes each element of $U_\Sigma(A)$ to it Q-equivalence class. □

2.8. Lemma. *Let $h: A \to B$ in $\mathbf{Alg}_\Sigma$, let $e_h: Q_h \to A \times A$ be the congruence on A induced by h, let $e: Q \to A \times A$ be another congruence on A such that $Q \subseteq Q_h$ in the precise sense that there exists a monomorphism $m: Q \to Q_h$ where $e_h \bullet m = e$, and let $g_Q: A \to A/Q$ be the homomorphism induced by Q. Then there exists a unique $\bar{g}: A/Q \to B$ such that $\bar{g} \bullet g_Q = h$.*

Proof. Because $e_h \bullet m = e$, we have

$$h \bullet p_{A,A} \bullet e = h \bullet p_{A,A} \bullet e_h \bullet m = h \bullet q_{A,A} \bullet e_h \bullet m = h \bullet q_{A,A} \bullet e.$$

Then, because $g_Q: A \to A/Q$ is a coequalizer for $p_{A,A} \bullet e$ and $q_{A,A} \bullet e$, it follows that there exists a unique $\bar{g}: A/Q \to B$ such that $\bar{g} \bullet g_Q = h$.

$$\begin{tikzcd} Q_h \arrow[dr, "e_h"] & & & & B \\ & A \times A \arrow[r, bend left, "p_{A,A}"] \arrow[r, bend right, "q_{A,A}"'] & A \arrow[urr, "h"] \arrow[drr, "g_Q"'] & & \\ Q \arrow[uu, "m"] \arrow[ur, "e"'] & & & & A/Q \arrow[uu, "\bar{g}"'] \end{tikzcd}$$

□

2.9. Proposition. *Let $h: A \to B$ in $\mathbf{Alg}_\Sigma$. Then there exists a surj-mono factorization of h, that is, there exists a Σ-algebra D, a surjective epimorphism $e: A \to D$, and a monomorphism $m: D \to B$ such that $m \bullet e = h$.*

$$\begin{tikzcd} A \arrow[rr, "h"] \arrow[dr, "e"'] & & B \\ & D \arrow[ur, "m"'] & \end{tikzcd}$$

Proof. Let $e_h: Q_h \to A \times A$ be the congruence on A induced by h. Let $D = A/Q_h$ and let $e: B \to A/Q_h$ be the coequalizer given by Proposition 2.7. By Proposition 2.8 we obtain a unique morphism $m: A/Q_h \to B$ such that $m \bullet e = h$; inspection of the proof of Proposition 2.7 shows that m will be injective. □

2.10. Lemma. *If $\beta \bullet \alpha$ is monic then α is monic.*

Proof. Let f, g be such that $\alpha \bullet f = \alpha \bullet g$, then we must show that $f = g$. But $\alpha \bullet f = \alpha \bullet g$ implies $\beta \bullet \alpha \bullet f = \beta \bullet \alpha \bullet g$ so $\beta \bullet \alpha$ monic implies $f = g$ as desired. $\square$

2.11. Proposition. *If A is a Σ-algebra and $\mathscr{B}$ is a subset of $U_\Sigma(A)$ then there exists a smallest subalgebra $B \subseteq A$ such that $\mathscr{B} \subseteq U_\Sigma(B)$.*

Proof. Let $\iota : \mathscr{B} \to U_\Sigma(A)$ be the inclusion mapping given by the inclusion $\mathscr{B} \subseteq U_\Sigma(A)$. Let $\iota^\# : F_\Sigma(\mathscr{B}) \to A$ be the unique homomorphism such that $U_\Sigma(\iota^\#) \bullet \eta_{\mathscr{B}} = \iota$. By Proposition 2.9, $\iota^\#$ has a surj-mono factorization $\langle e, m \rangle$

$$\begin{tikzcd}
F_\Sigma(\mathscr{B}) \arrow[dr, "e"] \arrow[dd, "\iota^\#"'] & \\
& A(\mathscr{B}) \arrow[dl, "m"] \\
A &
\end{tikzcd}$$

Now ι is monic, but

$$\iota = U_\Sigma(\iota^\#) \bullet \eta_{\mathscr{B}} = U_\Sigma(m) \bullet U_\Sigma(e) \bullet \eta_{\mathscr{B}},$$

so $U_\Sigma(m) \bullet U_\Sigma(e) \bullet \eta_{\mathscr{B}}$, is also monic, which, by Lemma 2.10, implies $U_\Sigma(e) \bullet \eta_{\mathscr{B}} : \mathscr{B} \to U_\Sigma(A(\mathscr{B}))$ is monic, i.e. $\mathscr{B} \subseteq U_\Sigma(A(\mathscr{B}))$.

To see that $A(\mathscr{B})$ is the least subalgebra of A containing $\mathscr{B}$, let us assume $k : C \to A$ is a subalgebra of A, and that $j : \mathscr{B} \to U_\Sigma(C)$ is monic in $\mathbf{Set}^S$ such that $U_\Sigma(k) \bullet j = \iota$, i.e. $\mathscr{B} \subseteq U_\Sigma(C) \subseteq U_\Sigma(A)$. Then what we wish to show is that $A(\mathscr{B})$ is a subalgebra of C.

Note first that

$$U_\Sigma(k \bullet j^\#) \bullet \eta_{\mathscr{B}} = U_\Sigma(k) \bullet U_\Sigma(j^\#) \bullet \eta_{\mathscr{B}} = U_\Sigma(k) \bullet j = \iota = U_\Sigma(\iota^\#) \bullet \eta_{\mathscr{B}}$$

implying that $k \bullet j^\# = \iota^\#$.

By construction, $e : F_\Sigma(\mathscr{B}) \to A(\mathscr{B})$ is a coequalizer; let it be the coequalizer of α and β. Then we have

$$k \bullet j^\# \bullet \alpha = \iota^\# \bullet \alpha = m \bullet e \bullet \alpha = m \bullet e \bullet \beta = \iota^\# \bullet \beta = k \bullet j^\# \bullet \beta.$$

But then k monic implies $j^\# \bullet \alpha = j^\# \bullet \beta$ so, since e is a coequalizer for α and β, there exists a unique $h : A(\mathscr{B}) \to C$ such that $h \bullet e = j^\#$. Then $k \bullet h \bullet e = k \bullet j^\# = m \bullet e$ so e epic yields $k \bullet h = m$, which, by Lemma 2.10, implies $h : A(\mathscr{B}) \to C$ is monic. $\square$

2.12. Proposition. *Let $\mathscr{Q} = \langle q_i : Q_i \to A \times A \mid i \in I \rangle$ be a nonempty family of congruences on A. Then there exists a congruence on A, denoted $\bigcap \mathscr{Q}$, such that $U_\Sigma(\bigcap \mathscr{Q}) = \bigcap \langle U_\Sigma(Q_i) \mid i \in I \rangle$. Equivalently, $\bigcap \mathscr{Q}$ is the limit of the evident diagram corresponding to $\mathscr{Q}$.*

2.13. Proposition. *Let $R \subseteq U_\Sigma(A \times A)$, then there exists a least congruence $Q(R)$ on A such that $R \subseteq U_\Sigma(Q(R))$.*

Proof. Let $\mathcal{Q}$ be the set of all congruences Q on A such that $R \subseteq U_\Sigma(Q)$. $\mathcal{Q}$ is nonempty since $1_{A\times A}: A \times A \to A \times A$ is in $\mathcal{Q}$. But clearly then $\bigcap \mathcal{Q}$ is the least congruence Q on A such that $R \subseteq U_\Sigma(Q)$. □

2.14. Definition. Let $P = \langle S, \Sigma, E\rangle$ be a presentation, and let $Y \in |\mathbf{Set}^S|$. Define *the E-relation for Y* to be the set of pairs

$$R(E, Y) = \{\langle \theta^{\#}(e_L), \theta^{\#}(e_R)\rangle \mid e = \langle e_L, e_R\rangle \in E, \text{and } \theta : \mathrm{var}(e) \to U_\Sigma(F_\Sigma(Y))\}.$$

2.15. Proposition. *Let $p = \langle S, \Sigma, E\rangle$ be a presentation. Given a P-algebra A let $\varepsilon_A : F_\Sigma(U_\Sigma(A)) \to A$ be the unique morphism given by the adjunction of Proposition 1.7 such that $U_\Sigma(\varepsilon_A) \bullet \eta_{U_\Sigma(A)} = 1_{U_\Sigma(A)}$. Define Q_A to be the congruence on $F_\Sigma(U_\Sigma(A))$ induced by ε_A. Then $A \cong F_\Sigma(U_\Sigma(A))/Q_A$, and $R(E, U_\Sigma(A)) \subseteq U_\Sigma(Q_A)$.*

Proof. From the fact that $U_\Sigma(\varepsilon_A) \bullet \eta_{U_\Sigma(A)} = l_{U_\Sigma(A)}$ in $\mathbf{Set}^S$, it follows that ε_A is surjective. By Proposition 2.9 we have a factorization $m \bullet e$ of ε_A through $F_\Sigma(U_\Sigma(A))/A_A$ with m injective. But ε_A surjective then implies m surjective, and so m is bijective, i.e. $m: A \cong F_\Sigma(U_\Sigma(A))/Q_A$. That $R(E, U_\Sigma(A)) \subseteq U_\Sigma(Q_A)$, follows from the definition of satisfaction, Definition 1.8. □

2.16. Proposition. *Let $P = \langle S, \Sigma, E\rangle$ be a presentation. Let $U_p : \mathbf{Alg}_P \to \mathbf{Set}^S$* be the restriction of U_Σ to $\mathbf{Alg}_P$. Then U_P has a left adjoint $F_P : \mathbf{Set}^S \to \mathbf{Alg}_P$. That is, for each $Y \in |\mathbf{Set}^S|$ there exists a P-algebra $F_P(Y)$, and a mapping $\eta_Y : U_P(F_P(Y))$ such that, for each P-algebra A, and each mapping $f: Y \to U_P$, there is a unique homomorphism $\bar{f}: F_P \to A$ such that $U_P(\bar{f}) \bullet \eta_Y = f$

$$\begin{tikzcd}
Y \arrow[r, "\eta_Y"] \arrow[dr, "f"'] & U_P(F_P(Y)) \arrow[d, "U_P(\bar{f})"] & F_P(Y) \arrow[d, "\bar{f}"] \\
& U_P(A) & A
\end{tikzcd}$$

Proof. By Proposition 1.7 there exists a free Σ-algebra, $F_\Sigma(Y)$, generated by Y, so, in particular, there exists a mapping $\bar{\eta}_Y : Y \to U_\Sigma(F_\Sigma(Y))$ such that for each Σ-algebra (and thus for each P-algebra) A, and mapping $f: Y \to U_\Sigma(A)$ there is a unique Σ-homomorphism $\hat{f}: F_\Sigma(Y) \to A$ such that $U_\Sigma(\hat{f}) \bullet \bar{\eta}_Y = f$. Now let $R(E, Y)$ be the E-relation for Y, and let Q_E be the smallest congruence on $F_\Sigma(Y)$ containing $R(E, Y)$ as given by Proposition 2.15. Then, by Proposition 2.6, there exists $Q(\hat{f})$, the congruence on $F_\Sigma(Y)$ induced by $\hat{f}$. Now assume that A is a P-algebra. This implies $R(E, Y) \subseteq U_\Sigma(Q(\hat{f}))$. But then, by Proposition 2.8, there exists a unique

$\bar{f}:(F_\Sigma(Y)/Q(E))\to A$ such that $\bar{f}\bullet k=\hat{f}$. It is easily checked then that the desired result follows by taking $|F_P|(Y)=F_\Sigma(Y)/Q(E)$, $\eta_Y=k\bullet\bar{\eta}_Y$.

$$\begin{tikzcd}
Y \arrow[r, "\bar{\eta}_Y"] \arrow[dr, "f"'] & U_\Sigma(F_\Sigma(Y)) \arrow[d, "U_\Sigma(\hat{f})"] & U_\Sigma(F_\Sigma(Y)/Q(E)) \arrow[l, "U_\Sigma(k)"'] \arrow[dl, "U_\Sigma(\bar{f})"] \\
 & U_\Sigma(A) &
\end{tikzcd}$$

$$\begin{tikzcd}
F_\Sigma(Y) \arrow[r, "k"] \arrow[d, "\hat{f}"] & F_\Sigma(Y)/Q(E) \arrow[dl, "\bar{f}"] \\
A &
\end{tikzcd}$$

□

2.17. Lemma. *Let $P=\langle S,\Sigma,E\rangle$ be a presentation. Let $X_1, X_2\in|\mathbf{Set}^S|$, and let $\iota_i:X_i\to X_1+X_2$, $i=1,2$, be a coproduct in* $\mathbf{Set}^S$, *then*

$$F_P(\iota_i):F_P(X_i)\to F_P(X_1+X_2)$$

is a coproduct in $\mathbf{Alg}_P$.

Proof. This is an immediate consequence of the fact that left adjoints preserve colimits. □

2.18. Proposition. $\mathbf{Alg}_P$ *has coproducts.*

Proof. *Let* A_1, $A_2\in|\mathbf{Alg}_P|$. Then, by Proposition 2.15, for $i=1,2$, there exists a congruence Q_i on $F_P(U_P(A_i))$, and a corresponding regular epimorphism $q_i:F_P(U_P(A_i))\to A_i\cong F_P(U_P(A_i))/Q_i$. By Lemma 2.17, we have a coproduct.

$$F_P(U_P(A_1))\xrightarrow{F_P(\iota_1)}F_P(U_P(A_1)+U_P(A_2))\xleftarrow{F_P(\iota_2)}F_P(U_P(A_2))$$

where $\iota_i:U_P(A_i)\to U_P(A_1)+U_P(A_2)$ is the indicated coproduct injection in $\mathbf{Set}^S$. Let Q be the smallest congruence on $F_P(U_P(A_1)+U_P(A_2))$ such that $U_P(Q)$ "contains" $U_P(Q_i)$, $i=1,2$, in the sense that if $\langle q_{i1},q_{i2}\rangle\in U_P(Q_i)$ then $\langle\iota_i(q_{i1}),\iota_i(q_{i2})\rangle\in U_P(Q)$. Let γ be the induced homomorphism

$$\gamma:F_P(U_P(A_1)+U_P(A_2))\to F_P(U_P(A_1)+U_P(A_2))/Q.$$

Clearly then, $U_P(Q_i)$ is contained in the congruence on $F_P(U_P(A_i))$ induced by $\gamma\bullet\iota_i$. But then, by Lemma 2.8, there exists a unique $\kappa_i:A_i\to F_P(U_P(A_1)+U_P(A_2))/Q$, $i=1,2$, such that $\kappa_i\bullet q_i=\gamma\bullet\iota_i$.

We claim that the desired coproduct is

$$A_1\xrightarrow{\kappa_1}F_P(U_P(A_1)+U_P(A_2))/Q\xleftarrow{\kappa_2}A_2.$$

To see that this is so let $f_i:A_i\to B$, $i=1,2$. Then we must show that there exists a unique $f:F_P(U_P(A_1)+U_P(A_2))/Q\to B$ such that $f\bullet\kappa_i=f_i$, $i=1,2$.

Now since $f_i \bullet q_i : F_P(U_P(A_i)) \to B$, $i = 1, 2$, it follows from Lemma 2.17, that there exists a unique $h : F_P(U_P(A_1) + U_P(A_2)) \to B$ such that

$$h \bullet F_P(\iota_i) = f_i \bullet q_i, \quad i = 1, 2.$$

But this implies that $Q(h)$, the congruence induced by h, contains $U_P(Q_i)$, $i = 1, 2$ (i.e. its image under ι_i), and so, by Lemma 2.8, there exists a unique $\bar{h} : F_P(U_P(A_1) + U_P(A_2)) \to B$ such that $\bar{h} \bullet \gamma = h$. But then

$$\bar{h} \bullet \kappa_i \bullet q_i = \bar{h} \bullet \gamma \bullet F_P(\iota_i) = h \bullet F_P(\iota_i) = f_i \bullet q_i$$

whence $\bar{h} \bullet \kappa_i = f_i$ since q_i is an epimorphism.

Now taking $f = \bar{h}$ will give the desired result providing $\bar{h}$ is unique. So assume $g : F_P(U_P(A_1) + U_P(A_2))/Q \to B$ such that $g \bullet \kappa_i = f_i$, $i = 1, 2$. Then $g \bullet \kappa_i = f_i$ implies $f_i \bullet q_i = g \bullet \kappa_i \bullet q_i = g \bullet \gamma \bullet F_P(\iota_i)$, implying $g \bullet \gamma = h$, which, in turn, implies $g = \bar{h}$ (cf. Fig. 2). □

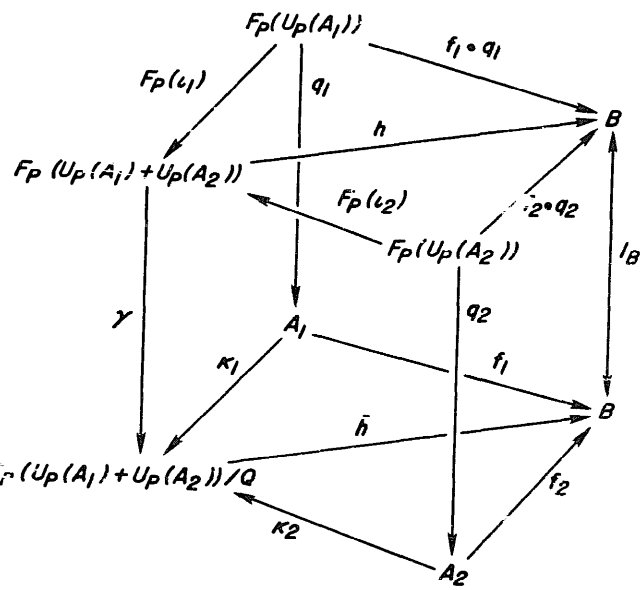

Fig. 2.

3. Polynomials

You are undoubtably familiar with the idea of a polynomial as something of the form

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0.$$

Many of you are also familiar with the somewhat more abstract notion of a polynomial over a ring. In the rest of this paper we will be investigating, and exploiting, the generalization of these ideas to arbitrary P-algebras. This generalization is "well known", and, indeed, what we do here is but a special case of the construction of what are called Kleisli categories (see [12]). What we have tried to do here is present this material in a way that facilitates its computer science application.

3.1. Definition. Let $P = \langle S, \Sigma, E\rangle$ be a presentation, and let $F_P \dashv U_P : \mathbf{Alg}_P \to \mathbf{Set}^S$ as in Proposition 2.16. Given $A \in |\mathbf{Alg}_P|$, and $X \in |\mathbf{Set}^S|$, define $A[X]$, *the algebra of polynomials in X over A*, to be the P-algebra $A + F_P(X)$. Define *the S-sorted set of polynomials in X over A to be* $U_P(A[X])$.

While it may not be immediately obvious, the familiar polynomials are just a special case of this definition. The familiar canonical form of polynomials over a ring is a reflection of the associative, commutative, and distributive laws for rings. Most of the cases of polynomials of computer science interest do not enjoy such nice canonical forms.

Before giving an example we note the following simple result.

3.2. Fact. *Let* $P = \langle S, \Sigma, E\rangle$ *be a presentation, and let* $X \in |\mathbf{Set}^S|$, *then*

$$\mathcal{I}nit_P[X] = \mathcal{I}nit_P + \mathcal{F}ree_P(X) = \mathcal{F}ree_P(X).$$

Applying these ideas to the presentation **Prim** yields a simple computer science example.

3.3. Example. Let **Prim** be the presentation given in Example 1.11, with $S = \{\mathrm{NAT}, \mathrm{BOOL}\}$, and let N and B be disjoint sets (of identifiers). Let $X \in |\mathbf{Set}^S|$ with $X_{\mathrm{NAT}} = N$, $X_{\mathrm{BOOL}} = B$. Then $\mathcal{I}nit_{\mathbf{Prim}}[X] = \mathcal{F}ree_{\mathbf{Prim}}(X)$ is the (S-sorted) set of arithmetic and boolean expressions in the operations from Σ and indicated identifiers.

Proposition 1.7 tells us how to evaluate the arithmetic and boolean expressions given values for the identifiers. The same idea is easily extended to polynomials over any P-algebra.

3.4. Definition. Let $P = \langle S, \Sigma, E\rangle$ be a presentation, A a P-algebra, and $X \in |\mathbf{Set}^S|$. Let $U_P : \mathbf{Alg}_P \to \mathbf{Set}^S$ be the underlying set functor for P-algebras. Then for any P-algebra B, any P-homomorphism $h : A \to B$, and $s \in S$, we have that

$$\mathbf{eval}_{h,s} : A[X]_s \times \mathbf{Set}^S(X, U_P(B)) \to U_P(B)_s,$$

$$\langle \alpha, f\rangle \mapsto (U_P[h, f^{\#}])_s(\alpha)$$

evaluates the polynomial α at (the argument) f, where $f^{\#} : F_P(X) \to B$ is the unique morphism such that $U_P(f^{\#}) \circ \eta_X = f$, and $[h, f^{\#}]$ is the indicated coproduct mediator

$$\begin{tikzcd} A \arrow[r, "\kappa_{A,F_P(X)}"] \arrow[dr, "h"'] & A[X] \arrow[d, "{[h,f^{\#}]}"] & F_P(X) \arrow[l, "\lambda_{A,F_P(X)}"'] \arrow[dl, "f^{\#}"] \\ & B & \end{tikzcd}$$

We can generalize the above in a simple, but very useful, manner. First a definition.

3.5. Definition. Let P be a presentation and let A be a P-algebra. Let $X, Y \in |\mathbf{Set}^S|$, then by *a Y-tuple of polynomials in A over X* we mean an S-sorted mapping $\alpha : Y \to U_P(A[X])$.

3.6. Definition. Let $P = \langle S, \Sigma, E \rangle$ be a presentation, A a P-algebra, and let $X, Y \in |\mathbf{Set}^S|$. Then for any P-algebra B, any P-homomorphism $h : A \to B$, and $s \in S$, we have

$$\mathbf{eval}_{X,Y,h} : \mathbf{Set}^S(X \to U_P(A[Y])) \times \mathbf{Set}^S(Y, U_P(B)) \to \mathbf{Set}^S(X \to U_P(B)),$$
$$\langle \alpha, f \rangle \mapsto [h, f^{\#}] \circ \alpha.$$

A special case of the above that is of special interest is when we take $B = A[Z]$ for some $Z \in |\mathbf{Set}^S|$, and take $h = \kappa_{A,F_P(Z)}$. In this case, it is quite reasonable to call the operation **subst** (for substitution) rather than as **eval** (for evaluation). In the case that A is a word algebra the operation gives exactly the substitution of terms in Z for the variables in Y, i.e. we simultaneously substitute $f(y)$ for y in each term $\alpha(x)$. The precise definition is as follows.

3.7. Definition. Given $P = \langle S, \Sigma, E \rangle$, $X, Y, Z \in |\mathbf{Set}^S|$, and $A \in \mathbf{Alg}_P$, define an *X-tuple of polynomials in Y over A*, to be an element of $\mathbf{Set}^S(X, U_P(A[Y]))$. Then define *general substitution in X, Y, Z over A* to be the operation

$$\mathbf{subst}_{X,Y} : \mathbf{Set}^S(X, U_P(A[Y])) \times \mathbf{Set}^S(Y, U_P(A[Z])) \to \mathbf{Set}^S(X, U_P(A[Z]))$$

such that, for $f : X \to U_P(A[Y])$, and $g : Y \to U_P(A[Z])$,

$$\mathbf{subst}(f, g) = (U_P([\kappa_{A,pF_P(Z)}, g^{\#}] \circ f^{\#})) \circ \eta_X.$$

Or, putting it the other way around, $\mathbf{subst}(f, g)$ is the unique morphism from X to $U_P(A[Z])$ such that $\mathbf{subst}(f, g)^{\#} = [\kappa_{A,F_P(Z)}, g^{\#}] \circ f^{\#}$

$$\begin{tikzcd} & A \arrow[d, "\kappa_{A,F_P(Y)}"'] \arrow[dr, "\kappa_{A,F_P(Z)}"] & \\ F_P(X) \arrow[r, "f^{\#}"] & A[Y] \arrow[r, "{[\kappa_{A,F_P(Z)}, g^{\#}]}"] & A[Z] \\ & F_P(Y) \arrow[u, "\lambda_{A,F_P(Y)}"] \arrow[ur, "g^{\#}"'] & \end{tikzcd}$$

One advantage of such an abstract definition of substitution is that it is independent of any particular representation of terms, and thus enables us to prove important properties of substitution in a representation independent manner. But, perhaps even more important, the proofs are very perspicuous. Here are some significant examples of proofs of properties of general substitution that will play an important role in the rest of our development.

3.8. Notation. From here on we shall abbreviate $\kappa_{A,F_P(X)}$ and $\lambda_{A,F_P(X)}$ respectively, to κ_X and λ_X respectively.

3.9. Proposition. *The substitution operation,* **subst**, *is associative. That is, given* $W, X, Y, Z \in |\mathbf{Set}^S|$, *and* $f: W \to U_P(A[X])$, $g: X \to U_P(A[Y])$, *and* $h: Y \to U_P(A[Z])$, *then*

$$\mathbf{subst}(\mathbf{subst}(f, g), h) = \mathbf{subst}(f, \mathbf{subst}(g, h)).$$

Proof. Because $F_P \dashv U_P$, it suffices to show that

$$\mathbf{subst}(\mathbf{subst}(f, g), h)^\# = \mathbf{subst}(f, \mathbf{subst}(g, h))^\#.$$

That may be done as follows:

$$\begin{aligned}\mathbf{subst}(\mathbf{subst}(f, g), h)^\# &= [\kappa_Z, h^\#] \bullet \mathbf{subst}(f, g)^\# = [\kappa_Z, h^\#] \bullet [\kappa_Y, g^\#] \bullet f^\# \\ &= [[\kappa_Z, h^\#] \bullet \kappa_Y, [\kappa_Z, h^\#] \bullet g^\#] \bullet f^\# \\ &= [\kappa_Z, [\kappa_Z, h^\#] \bullet g^\#] \bullet f^\# = [\kappa_Z, \mathbf{subst}(g, h)^\#] \bullet f^\# \\ &= \mathbf{subst}(f, \mathbf{subst}(g, h))^\#.\end{aligned}$$

$$\begin{tikzcd}
& A \arrow[d, "\kappa_X"'] \arrow[dr, "\kappa_Y"] & A \arrow[d, "\kappa_Y"] \arrow[dr, "\kappa_Z"] & \\
F_P(W) \arrow[r, "f^\#"] & A[X] \arrow[r, "{[\kappa_Y, g^\#]}"] & A[Y] \arrow[r, "{[\kappa_Z, h^\#]}"] & A[Z] \\
& F_P(X) \arrow[u, "\lambda_X"] \arrow[ur, "g^\#"'] & F_P(Y) \arrow[u, "\lambda_Y"'] \arrow[ur, "h^\#"'] &
\end{tikzcd}$$

□

3.10. Proposition. *Let* $X \in |\mathbf{Set}^S|$, *and let* $\mathrm{id}_X = U_P(\lambda_X) \bullet \eta_X : X \to U_P(A[X])$ (*or, equivalently, let* $(\mathrm{id}_X)^\# = \lambda_X$).

$$\begin{tikzcd}
X \arrow[r, "\eta_X"] \arrow[dr, "\mathrm{id}_X"'] & U_P(F_P(X)) \arrow[d, "U_P(\lambda_X)"] \\
& U_P(A[X])
\end{tikzcd}
\qquad
\begin{tikzcd}
F_P(X) \arrow[d, "\lambda_X"] \\
A[X]
\end{tikzcd}$$

Then id_X *is the identity on* $\mathbf{Set}^S(X, U_P(A[X]))$ *with respect to* **subst**, *i.e. for any Y-tuple of polynomials* $\alpha: Y \to U_P(A[X])$, $\mathbf{subst}(\alpha, \mathrm{id}_X) = \alpha$, *and for any X-tuple of polynomials* $\beta: X \to U_P(A[Y])$, $\mathbf{subst}(\mathrm{id}_X, \beta) = \beta$.

Proof. Again, because of the adjunction $F \dashv U$, it suffices to show that

$$\mathbf{subst}(\alpha, \mathrm{id}_X)^{\#} = a^{\#} \quad \text{and} \quad \mathbf{subst}(\mathrm{id}_X, \beta)^{\#} = \beta^{\#}.$$

But this is easily seen as follows:

$$\mathbf{subst}(\alpha, \mathrm{id}_X)^{\#} = [\kappa_X, (\mathrm{id}_X)^{\#}] \bullet \alpha^{\#} = [\kappa_X, \lambda_X] \bullet \alpha^{\#} = \alpha^{\#}.$$

While, on the other hand,

$$\mathbf{subst}(\mathrm{id}_X, \beta)^{\#} = [\kappa_Y, \beta^{\#}] \bullet \lambda_X = \beta^{\#}. \qquad \square$$

The associativity of **subst** together with the existence of identities shows that for any collection $\mathcal{S}$ of S-sorted sets and any P-algebra A, we can form a category "of polynomials" in which the morphisms from X to Y are Y-tuples (i.e. Y-index sets) of polynomials in X over A.

3.11. Definition. Let $\mathcal{S}$ be an S-ary subset of $|\mathbf{Set}^S|$, then we define $\mathbf{Poly}_{A,\mathcal{S}}$, *the category of polynomials in A and $\mathcal{S}$*, with $|\mathbf{Poly}_{A,\mathcal{S}}| = \mathcal{S}$ and for X, $Y \in \mathcal{S}$, $\mathbf{Poly}_{A,\mathcal{S}}(X, Y)$ consists of all triples

$$\alpha = \langle X, \bar{\alpha} : Y \to U_P(A[X]), Y\rangle.$$

Note that $\alpha : X \to Y$, but $\bar{\alpha}$, the underlying tuple of polynomials, is a mapping $\bar{\alpha} : Y \to U_P(A[X])$. Given $\alpha \in \mathbf{Poly}_{A,\mathcal{S}}(X, Y)$ and $\beta \in \mathbf{Poly}_{A,\mathcal{S}}(Y, Z)$, we define their composite, $\beta \circ \alpha \in \mathbf{Poly}_{A,\mathcal{S}}(X, Z)$ to be the triple

$$\beta \circ \alpha = \langle X, \mathbf{subst}(\alpha, \beta), Y\rangle.$$

Given $X \in \mathcal{S}$, define the identity on X to be the triple,

$$1_X = \langle X, \mathrm{id}_X, X\rangle$$

where $\mathrm{id}_X = U_P(\lambda_X) \bullet \eta_X : X \to U_P(A[X])$, as in Proposition 3.10.

3.12. Example. Let I be a set of integer identifiers and let B be a set of boolean identifiers. Let $S = \{\mathrm{INT}, \mathrm{BOOL}\}$ and let $\mathcal{S}$ be the S-sorted set Id with $\mathrm{Id}(\mathrm{INT}) = N$, and $\mathrm{Id}(\mathrm{BOOL}) = B$. Let **Prim** be the presentation given in Example 1.11, let **FPr** be the free **Prim**-algebra and let **IPr** be the "usual interpretation" for **Prim**. Each Id-tuple of polynomials α in $\mathbf{Poly}_{\mathbf{FPr},\mathcal{S}}$ associates an arithmetic expression with each integer identifier in I, and associates a boolean expression with each boolean identifier in B. Thus, syntactically, each such Id-tuple looks like simultaneous assignment statement. This is not a meaningless coincidence. The intuitive semantics of such a simultaneous assignment is that it is a state-to-state transformation, where a state is an assignment of values to identifiers. In particular, given a state and a simultaneous assignment, the desired transformation results from replacing the value of each identifier by the value of the corresponding expression evaluated using the values of the identifiers in the given state. But this is easily restated precisely as follows. Define a *state* as an S-sorted mapping $\sigma : \mathrm{Id} \to U_{\mathbf{Prim}}\mathbf{IPr}[\mathrm{Id}])$. The desired semantics for the elements of $\mathbf{Poly}_{\mathbf{FPr},\mathcal{S}}$ with respect to the interpretation $In : \mathbf{FPr} \to \mathbf{IPr}$ is exactly that given by

$$\mathbf{eval}_{\mathrm{Id},\mathrm{Id},In} : \mathbf{Set}^S(\mathrm{Id}, U_P(\mathbf{FPr}[\mathrm{Id}])) \to \mathbf{Set}^S(\mathrm{Id}, U_P(In)).$$
$$\langle \alpha, \sigma\rangle \mapsto U_P([In, \sigma^{\#}]) \bullet \alpha.$$

If we choose $\mathcal{S}$ so that it is closed under finite coproducts, then $\mathbf{Poly}_{A,\mathcal{S}}$ will have finite products.

3.13. Fact. *For any set S, $\mathbf{Set}^S$ has coproducts, indeed if X, $Y \in |\mathbf{Set}^S|$, then $X + Y = \langle X_s + Y_s \mid s \in S$, $+$ as in $\mathbf{Set}\rangle$, that is, $(X+Y)_s = X_s + Y_s$, and similarly with the coproduct injections.*

3.14. Proposition. *Let $\mathcal{S}$ be an S-ary subset of $|\mathbf{Set}^S|$. Then, for any two S-ary sets X and Y in $\mathcal{S}$ and coproduct*

$$\langle \kappa_{X,Y} : X \to X + Y, \lambda_{X,Y} : Y \to X + Y\rangle$$

in $\mathbf{Set}^S$, we have a product object $X \times Y$ in $\mathbf{Poly}_{A,\mathcal{S}}$, where $X \times Y = A[X+Y]$ *and we have product projections*

$$p_{X,Y} : X \times Y \to X \quad \textit{and} \quad q_{X,Y} : X \times Y \to Y$$

in $\mathbf{Poly}_{A,\mathcal{S}}$, where

$$(\bar{p}_{X,Y})^{\#} = \lambda_{X+Y} \bullet F_P(\kappa_{X,Y}) \quad \textit{and} \quad (\bar{q}_{X,Y})^{\#} = \lambda_{X+Y} \bullet F_P(\lambda_{X,Y}).$$

$$\begin{tikzcd}
F_P(X) \arrow[dr, "F_P(\kappa_{X,Y})"'] \arrow[drr, "(\bar{p}_{X,Y})^{\#}"] & & \\
& F_P(X+Y) \arrow[r, "\lambda_{X+Y}"] & A[X+Y] \\
F_P(Y) \arrow[ur, "F_P(\lambda_{X,Y})"] \arrow[urr, "(\bar{q}_{X,Y})^{\#}"'] & &
\end{tikzcd}$$

In particular, if $\alpha : X \to Z$ and $\beta : Y \to Z$ in $\mathbf{Poly}_{A,\mathcal{S}}$, then the desired mediating morphism $\langle \alpha, \beta\rangle$ in $\mathbf{Poly}_{A,\mathcal{S}}(Z, X \times Y)$ is given by the morphism h such that $\bar{h}^{\#} = [\bar{\alpha}^{\#}, \bar{\beta}^{\#}]$.

$$\begin{tikzcd}
F_P(X) \arrow[d, "F_P(\kappa_{X,Y})"'] \arrow[dr, "\bar{\alpha}^{\#}"] & \\
F_P(X+Y) \arrow[r, "{[\bar{\alpha}^{\#}, \bar{\beta}^{\#}]}"] & A[Z] \\
F_P(Y) \arrow[u, "F_P(\sigma_{X,Y})"] \arrow[ur, "\bar{\beta}^{\#}"'] &
\end{tikzcd}$$

Proof. Since $F_P \dashv U_P : \mathbf{Alg}_P \to \mathbf{Set}^S$, and left adjoints preserve colimits, it follows that F_P preserves coproducts and thus that

$$\langle F_P(\kappa_{X,Y}) : F_P(X) \to F_P(X+Y), F_P(\lambda_{X,Y}) : F_P(Y) \to F_P(X+Y) \rangle$$

is a coproduct in $\mathbf{Alg}_P$.

By definition, $p_{X,Y} \circ \langle \alpha, \beta \rangle = \mathbf{subst}(\bar{p}_{X,Y}, \langle \alpha, \beta \rangle^{-})$. But,

$$\begin{aligned}\mathbf{subst}(\bar{p}_{X,Y}, \langle \alpha, \beta \rangle^{-})^{\#} &= [\kappa_Z, \langle \alpha, \beta \rangle^{-\#}] \bullet (\bar{p}_{X,Y})^{\#} \\ &= [\kappa_Z, [\bar{\alpha}^{\#}, \bar{\beta}^{\#}]] \bullet \lambda_{X+Y} \bullet F_P(\kappa_{X,Y}) \\ &= [\bar{\alpha}^{\#}, \bar{\beta}^{\#}] \bullet F_P(\kappa_{X,Y}) \\ &= \bar{\alpha}^{\#}.\end{aligned}$$

So $p_{X,Y} \circ \langle \alpha, \beta \rangle = \alpha$ as required. A similar argument yields $q_{X,Y} \circ \langle \alpha, \beta \rangle = \beta$. The uniqueness of $\langle \alpha, \beta \rangle$ is easily shown. □

4. Lawvere algebraic theories

For this part of the paper it will be convenient to restrict our choice of $\mathcal{S} \subseteq |\mathbf{Set}^S|$ to a case where the S-ary set $\mathcal{S}$ is indexed, in a natural way, by the set, S^*, of strings on S. Part of the "naturalness" arises from S^* forming the set of objects of a category $\mathbf{Str}_S$ with coproducts corresponding to the concatenation of strings. The use of $\mathbf{Str}_S$ provides a convenient notation, and reflects some of the important underlying mathematical structure.

4.1. Definition. Given a set S, let S^*, the *set of strings on* S, be the set of all mappings $w:[n] \to S$, $n \in \omega$. A string $u:[n] \to S$ is said to be of *length* n. Let $\varepsilon:[0] \to S$ denote the unique mapping from $[0]$ to S; ε is called *the empty string.*

4.2. Example. We form a category $\mathbf{Str}_S$ with objects $|\mathbf{Str}_S| = S^*$, where for $w, u \in |\mathbf{Str}_S|$, with $w:[n] \to S$, and $u:[p] \to S$, a *string morphism* $f: w \to u$ is a mapping $f:[n] \to [p]$ such that $u \bullet f = w$, i.e. such that the following diagram commutes:

$$\begin{tikzcd} {[n]} \arrow[rr, "f"] \arrow[dr, "w"'] & & {[p]} \arrow[dl, "u"] \\ & S & \end{tikzcd}$$

4.3. Fact. *The category* $\mathbf{Str}_S$ *has coproducts. In particular, given any string* $u = u_1 \ldots u_n$, *the coproduct injection* $i_u : u_i \to u$ *is the string morphism given by the mapping* $i:[1] \to [n]$, $1 \mapsto i$, *and so the family* $\langle i_u : u_i \to u \mid i \in [b] \rangle$ *is a coproduct in* $\mathbf{Str}$.

Given S, let $\mathbf{Str}_S$ be the category of strings on S. Define $\rho:\mathbf{Str}_S \to \mathbf{Set}^S$ to be the functor such that, for each $u = u_1 \dots u_n \in |\mathbf{Str}_S|$

$$\rho(u) = \langle \rho(u)_s = \{x_i^u \mid i \in [n], u_i = s\} \mid s \in S\rangle$$

and such that for each $f: u \to v$

$$\rho(f) = \langle \rho(f)_s : \rho(u)_s \to \rho(v)_s \mid s \in S\rangle$$

where, if $x_i^u \in \rho(u)_s$ then $(\rho(u)_s)(x_i^u) = x_{f(i)}^v$.

4.4. Fact. *The functor $\rho:\mathbf{Str}_S \to \mathbf{Set}^S$ preserves coproducts. Thus, by essentially the same argument as in Proposition 3.14, we see that each coproduct injection $i_u : u_i \to u$ in $\mathbf{Str}_S$ induces a product projection, $X_i^u : A[u] \to A[u_i]$ in $\mathbf{Poly}_{A,\mathscr{S}}$, where $\mathscr{S} = |\mathbf{Str}_S|$.*

Proof. From the above we know that the family $\langle i_u : u_i \to u \mid i \in [n]\rangle$ is a coproduct for $u = u_1 \dots u_n \in |\mathbf{Str}_S|$. Now $\rho(u_i) = \{x_1^{u_i}\}$, and $\rho(u) = \{x_1^u, x_2^u, \dots, x_n^u\}$, and, for each $i \in [n]$

$$\rho(i_u) : \{x_1^{u_i}\} \to \{x_1^u, \dots, x_n^u\}, \qquad x_1^{u_i} \mapsto x_i^u,$$

from which the desired result follows by inspection. $\square$

4.5. Notation. Taking advantage of the above defined functor, ρ, we will restrict our attention to polynomials over $|\mathbf{Str}_S|$. For all $v \in S^*$ we will generally abbreviate $\rho(v)$ to v in expressions, so, for example, we will write $A[u]$ for $A[\rho(u)]$, $A[u+v]$ for $A[\rho(u)+\rho(v)]$, $F_P(v)$ for $F_P(\rho(v))$, κ_v for $\kappa_{\rho(v)}$, etc.

From here on we will assume that $\boldsymbol{P} = \langle S, \Sigma, E\rangle$ is a fixed presentation, that A is a fixed $\boldsymbol{P}$-algebra, and that $\mathscr{S} = |\mathbf{Str}_S| = S^*$. We shall write $\mathbf{Poly}_A$ for the category of polynomials $\mathbf{Poly}_{A,\mathscr{S}}$.

Let η^Σ denote the unit of the adjunction $F_\Sigma \dashv U_\Sigma : \mathbf{Alg}_\Sigma \to \mathbf{Set}^S$, and let η^P denote the unit of the adjunction $F_P \dashv U_P : \mathbf{Alg}_P \to \mathbf{Set}^S$.

For $v \in S^*$ let τ_v denote the unique homomorphism

$$\tau_v : F_\Sigma(v) \to (A + F_P(v))$$

such that the following diagram commutes:

$$\begin{tikzcd}
\rho(v) \arrow[r, "(\eta^\Sigma)_v"] \arrow[d, "(\eta^P)_v"'] & U_\Sigma(F_\Sigma(v)) \arrow[d, "U_\Sigma(\tau_v)"] \\
U_P(F_P(v)) \arrow[r, "U_P(\kappa_v)"'] & U_P(A + F_P(v))
\end{tikzcd}$$

If we define F_Σ as in Definition 1.6 then $U_\Sigma(F_\Sigma(v))$ is an S-ary set of terms written in the signature $\Sigma(v)$. We can use tuples of such terms to represent certain morphisms in $\mathbf{Poly}_A$. If $v = v_1 \dots v_n$, and $w = w_1 \dots w_p \in S^*$, then a morphism

$$\alpha : \rho(w) \to U_\Sigma(F_\Sigma(v))$$

can be written as a p-tuple of terms

$$\langle \alpha(x_1^u), \alpha(x_2^u), \ldots, \alpha(x_p^u)\rangle$$

and then regarded as representing the morphism $\bar{\alpha}$ in $\mathbf{Poly}_A(v, w)$ such that

$$\bar{\alpha} : \rho(w) \to U_P(A + F_P(v)), \qquad x_i^w \mapsto \tau_v(\alpha(x_i^w)).$$

Using this notation we see immediately that for w as above,

$$\langle x_1^w, x_1^w, \ldots x_p^w\rangle : w \to w$$

is the identity on w, and the family

$$\langle\langle x_i^w\rangle : w \to w_i \mid i = 1, \ldots, p\rangle$$

is a product (see Proposition 4.4 above).

The above remarks should provide some motivation for the following definition of an S-sorted algebraic theory.

4.6. Definition. Let S be a set, then an *S-sorted algebraic theory* is an $S^* \times S^*$-sorted algebra with

Carriers: $T(u, v)$, $u, v \in S^*$, and

Operations:

(a) $x_i^u \in T(u, u_i)$, for each $u = u_1 \ldots u_n \in S^*$ and i, $1 < i \leq n$,

(b) $\bullet_{u,v,w} : T(u, v) \times T(v, w) \to T(u, w)$, for all $u, v, w \in S^*$,

(c) $\langle, \ldots, \rangle_{v,u} : T(v, u_1) \times \cdots \times T(v, u_n) \to T(v, u)$, for all $v \in S^*$ and $u = u_1 \ldots u_n \in S^*$.

Axioms:

(4.6.1) $x_i^u \bullet_{v,u,u_i} \langle \alpha_1 \cdots \alpha_n\rangle = \alpha_i (\alpha_i \in T(v, u_i), 1 \leq i \leq n)$,

(4.6.2) $\langle x_1^u \bullet_{u_1,u,v} \beta, \ldots, x_n^u \bullet_{u_n,u,v} \beta\rangle_{u,v} = \beta\ (\beta \in T(v, u))$,

(4.6.3) $\langle x_1^u\rangle_{u_1,v} = x_1^u\ (u \in S^*)$,

(4.6.4) $(\gamma \bullet_{t,u,v} \beta) \bullet_{t,u,v} \alpha = \gamma \bullet_{u,v,w} (\beta \bullet_{t,u,w} \alpha)\ (\alpha \in T(t, u), \beta \in T(u, v), \gamma \in T(v, w))$,

(4.6.5) Let $1_u = \langle x_1^u, x_2^u, \ldots, x_n^u\rangle_{u,u} \in T(u, u)$, $u \in S^n$, then $1_u \bullet_{v,u,u} \beta = \beta$ for all $\beta \in T(v, u)$, and $\gamma \bullet_{u,u,w} 1_u = \gamma$ for all $\gamma \in T(u, w)$.

Elements of $T(u, v)$ are called *morphisms* with *source u* and *target v*. The operations x_i^u are often called *distinguished morphisms*, the operations $\bullet_{u,v,w}$ are called *composition operations*, the operations $\langle, \ldots, \rangle_{u,v}$ are called *tupling operations*. Operations $f : u \to w$ built up by tupling distinguished morphisms are called *base morphisms*. When there is no ambiguity we write "$\bullet$" rather than "$\bullet_{u,v,w}$", and "$\langle, \ldots, \rangle$" rather than "$\langle, \ldots, \rangle_{u,v}$".

If S is a singleton set, then we say that T is a 1-*sorted algebraic theory*, and we identify the elements of S^* with the natural numbers, i.e. where $S = \{s\}$, we write n for s^n.

Let T_1 and T_2 by S-sorted algebraic theories, then by an *S-sorted theory morphism* $H: T_1 \to T_2$, we mean an $(S^* \times S^*)$-indexed family of mappings $H = \langle H_{u,v}: T_1(u, v) \to T_2(u, v) \mid u, v \in S^* \rangle$ such that

(1) for each $u = u_1 \ldots u_n \in S^*$ and $i \in [n]$,

$$H_{u,u_i}(x_i^u) = x_i^u;$$

(2) if $\alpha: u \to v$, and $\beta: w \to u$ then

$$H_{w,v}(\alpha \bullet \beta) = H_{u,v}(\alpha) \bullet H_{w,v}(\beta);$$

(3) if $\alpha_i: u \to v_i$, $i = 1, \ldots, n$, then

$$H_{u,v}(\langle \alpha_1, \ldots, \alpha_n \rangle) = \langle H_{u,v_1}(\alpha_1), \ldots, H_{u,v_n}(\alpha_n) \rangle.$$

Let $\mathbf{Th}_S$ denote the category whose objects are the S-sorted theories and whose morphisms are the S-sorted theory morphisms. We call $\mathbf{Th}_S$ *the category of S-sorted theories.*

4.7. Theorem. $\mathbf{Poly}_A$ *is an S-sorted algebraic theory in the specific sense that there exists an S-sorted algebraic theory T_A with composition $\bullet$, such that*

(1) $T_A(u, v) = \mathbf{Poly}_A(u, v)$ *for all* $u, v \in S^*$,

(2) *if* $\alpha \in T_A(u, v)$ *and* $\beta \in T(v, w)$ *then* $\beta \bullet \alpha = \beta \circ \alpha$;

(3) *if* $u: [n] \to S$ *and* $\alpha_i \in T_A(u(i), v)$ *for each* $i \in \{n\}$, *then we define* $\langle \alpha_1, \ldots, \alpha_n \rangle$ *in* T_A *to be the morphism* $\langle \alpha_1, \ldots, \alpha_n \rangle$ *from* $\mathbf{Poly}_A$;

(4) *given* $u = u_1 \ldots u_n \in S^*$ *and* $i \in [n]$, *we define the distinguished morphism* $x_i^u \in T_A(u, u_i)$ *to be the product projection in* $\mathbf{Poly}_A(u, u_i)$ *corresponding to the coproduct injection* $i_u: u_i \to u$ *in* $\mathbf{Str}_S$ *(see above).*

Proof. This is a straightforward matter of showing that the axioms given in Definition 4.6 are satisfied. □

4.8. Definition. Let T be an S-sorted algebraic theory. Define a *T-algebra* to be a product preserving functor $\mathscr{A}: T \to \mathbf{Set}$. It will simplify our development to assume that products are preserved canonically in the sense that if $w = w_1 \ldots w_n \in S^*$ then $\mathscr{A}(w) = \mathscr{A}(w_1) \times \cdots \times \mathscr{A}(w_n)$ using a fixed notion of n-ary Cartesian product for all T-algebras. This eliminates having to deal, in the following proofs, with T-algebras which are isomorphic, but distinct, yet really only differ in the choice of products.

Given two T-algebras $\mathscr{A}$ and $\mathscr{B}$ then a *T-homomorphism* $h: \mathscr{A} \to \mathscr{B}$ is a natural transformation from $\mathscr{A}$ to $\mathscr{B}$.

For any algebraic theory T, let $\mathbf{Alg}_T$ denote the category with T-algebras as objects and T-homomorphisms as morphisms.

Let A be an S-sorted Σ-algebra and let $T_A = \mathbf{Poly}_A$ be the corresponding algebraic theory. Then define θ to be the $\mathbf{Set}^{S^* \times S}$-morphism

$$\theta: \Sigma \to \langle T_A(u, s) \mid u \in S^*, s \in S \rangle,$$

where, for $\sigma \in \Sigma_{w,s}$, $w = w_1 \dots w_p$, we have

$$\theta_{w,s}(\sigma) = \langle \sigma(x_1^w \dots x_p^w)\rangle$$

using the notational conventions given in Notation 4.5, so that $\langle \sigma(x_1^w \cdots x_p^w)\rangle$ is the map $\{x_1^s\}$ to $A[w]$, taking x_1^s to the indicated element of $A[w]$.

4.9. Proposition. *Let A be a $\boldsymbol{P}$-algebra. Then every $\boldsymbol{T}_A$-algebra is a $\boldsymbol{P}$-algebra in the sense that there is an embedding of the category $\mathbf{Alg}_{T_A}$ of $\boldsymbol{T}_A$-algebras into the category $\mathbf{Alg}_P$ of $\boldsymbol{P}$-algebras given as follows. If $\mathscr{A} : \boldsymbol{T}_A \to \mathbf{Set}$ is a $\boldsymbol{T}_A$-algebra then $\mathscr{A}$ induces an Σ-algebra, denoted A, where $A_s = \mathscr{A}(s)$ for every $s \in S$, and $\sigma_A = \mathscr{A}(\theta(\sigma)) = \mathscr{A}(\langle \sigma(x_1^w \dots x_p^w)\rangle)$. Furthermore, if $\mathscr{A}, \mathscr{B} : \boldsymbol{T}_A \to \mathbf{Set}$ are $\boldsymbol{T}_A$-algebras, and $h : \mathscr{A} \to \mathscr{B}$ is a $\boldsymbol{T}_A$-homomorphism (i.e. a natural transformation between the two functors), then the family*

$$h = \langle h_s : A_s \to B_s \mid s \in S\rangle$$

with h_s being the s-component $h_s : \mathscr{A}(s) \to \mathscr{B}(s)$ of the natural transformation, is the corresponding $\boldsymbol{P}$-homomorphism.

Proof. Since $\langle \sigma(x_1^w \dots x_p^w)\rangle \in \mathbf{Poly}_A(w, s)$, and $\mathscr{A}$ preserves products canonically, it follows that $\sigma_A = \mathscr{A}(\langle \sigma(x_1^w \dots x_p^w)\rangle) : \mathscr{A}_{w_1} \times \cdots \times A_{w_p} \to A_s$, and so A is a Σ-algebra. It remains to show that A is a $\boldsymbol{P}$-algebra, i.e. that it satisfies E. So, let $e = \langle L, R\rangle \in E$, where, say, $L, R \subset (\mathscr{F}\!\mathit{ree}_\Sigma(w))_s$ for some $s \in S$ and some $w \in S^*$. It is no loss of generality to assume that $\mathrm{var}(e) = \rho(w)$. We must show, for each S-ary mapping $f : \rho(w) \to U_\Sigma(A)$, that $f^\#(L) = f^\#(R)$ where $f^\# : F_\Sigma(w) \to A$ is the unique homomorphism such that $U_\Sigma(f^\#) \bullet (\eta^\Sigma)_w = f$. Now, since $A + F_p(w)$ is a $\boldsymbol{P}$-algebra, we already know that $\tau_w(R) = \tau_w(L)$ (τ_w as in Notation 4.5). Thus it will suffice to show that, where we define

$$\hat{f}(t) = (\mathscr{A}(\tau_w(t))(f(x_1^w), \dots, f(x_p^w))$$

for each $t \in U_\Sigma(F_\Sigma(w))$, that $\hat{f}$ is $f^\#$. Now, for any $w = w_1 \dots w_p \in S^*$ and $i \in [p]$,

$$\begin{aligned}\hat{f}(x_i^w) &= (\mathscr{A}(\tau_w(t))(f(x_1^w), \dots, f(x_p^w))\\ &= (\mathscr{A}(\langle x_i^w\rangle))(f(x_1^w), \dots, f(x_p^w))\\ &= f(x_i^w)\end{aligned}$$

since $\langle x_i^w\rangle : w \to w_i$ is a product projection and $\mathscr{A}$ preserves products. Thus $U_\Sigma(\hat{f}) \bullet (\eta^\Sigma)_w = f$. It remains then to show that $\hat{f}$ is a homomorphism, i.e. that if $\sigma \in \Sigma_{u,s}$ and $t_1, \dots, t_p \in U_\Sigma(F_\Sigma(w))$, then $\hat{f}(\sigma(t_1 \dots t_p)) = \sigma_A(\hat{f}(t_1), \dots, \hat{f}(t_p))$. But

$$\begin{aligned}\hat{f}(\sigma(t_1 \cdots t_p)) &= \mathscr{A}(\langle \sigma(t_1 \dots t_p)\rangle)(f(x_1^w), \dots, f(x_p^w))\\ &= \mathscr{A}(\langle \sigma(x_1^u \dots x_n^u)\rangle \bullet \langle t_1, \dots, t_n\rangle)(f(x_1^w), \dots, f(x_p^w))\\ &= (\mathscr{A}(\langle \sigma(x_1^u \dots x_n^u)\rangle) \bullet (\mathscr{A}(\langle t_1, \dots, t_n\rangle)) \bullet (f(x_1^w), \dots, f(x_p^w))\\ &= \sigma_A(\mathscr{A}(t_1), \dots, \mathscr{A}(t_n))(f(x_1^w), \dots, f(x_p^w))\\ &= \sigma_A(\mathscr{A}(t_1)(f(x_1^w), \dots, f(x_p^w)), \dots, \mathscr{A}(t_n)(f(x_1^w), \dots, f(x_p^w)))\\ &= \sigma_A(\hat{f}(t_1), \dots, \hat{f}(t_n)),\end{aligned}$$

so $\hat{f}$ is a homomorphism, and, by the uniqueness of $f^\#$, we have $\hat{f} = f^\#$. $\square$

4.10. Proposition. *Let C be a category with small hom sets (i.e. $C(X, Y)$ is always a set), let $G: C \to \mathbf{Set}$ be a functor, let $X \in |C|$, and let $\mathbf{Nat}(C(X,\_), G)$ denote the set of natural transformations from the hom-functor $C(X,\_)$ to G. Then there is a bijection*

$$\varphi : \mathbf{Nat}(C(X,\_), G) \cong G(X), \qquad \eta \mapsto \eta_X(1_X).$$

4.11. Proposition. *Let T be an algebraic theory. Then the hom-functor $T(\varepsilon,\_)$ is an initial T-algebra. And for each $w \in S^*$, the hom-functor $T(w,\_)$ is the T-algebra freely generated by $\rho(w)$.*

Proof. Let $\mathscr{A}: T \to \mathbf{Set}$ be any T-algebra. By Yoneda's Lemma $\mathbf{Nat}(T(\varepsilon,\_), \mathscr{A}) \cong \mathscr{A}(\varepsilon)$. But $\mathscr{A}(\varepsilon)$ is necessarily a singleton set and so there is exactly one T-homomorphism from $T(\varepsilon,\_)$ to $\mathscr{A}$. But that is exactly the definition of $T(\varepsilon,\_)$ being initial.

Let $w = w_1 \dots w_p$. By Yoneda's Lemma, we have $\phi : \mathbf{Nat}(T(w,\_), \mathscr{A}) \cong \mathscr{A}(w) = \mathscr{A}(w_1) \times \cdots \times \mathscr{A}(w_p)$, where, if $h : T(w,\_) \to \mathscr{A}$, then $\phi(h) = h_w(1_w) = (h(x_1^w), \dots, h(x_n^w))$, i.e. h is completely determined by its values on $\rho(w)$. □

4.12. Proposition. *Let A be a Σ-algebra, and let T_A be the algebraic theory corresponding to $\mathbf{Poly}_A$, then A is an initial T_A-algebra.*

Proof. By definition, $\mathbf{Poly}(\varepsilon,\_)(s) = \mathbf{Poly}(\varepsilon, s) = \mathbf{Set}^S(\{x_1^s\}, U_P(A + F_P(\varepsilon)))$. But $A + F_P(\varepsilon) = A$, so $\mathbf{Set}^S(\{x_1^s\}, U_P(A + F_P(\varepsilon))) = A_s$. Now given $\sigma \in \Sigma_{u,s}$, $u = u_1 \dots u_n$, and $(a_1, \dots, a_n) \in A_{u_1} \times \cdots \times A_{u_n}$, we see that

$$\begin{aligned}(\mathbf{Poly}(\varepsilon,\_)(\sigma))(a_1, \dots, a_n) &= (a_1, \dots, a_n)^{\#} \bullet \langle \sigma(x_1^u \dots x_n^u)\rangle \\ &= \sigma_A(a_1, \dots, a_n),\end{aligned}$$

$$\begin{tikzcd}
 & A \arrow[d] \arrow[dr, "1_A"] & \\
F_P(s) \arrow[r, "\sigma(x_1^u \dots x_n^u)"] & A + F_P(u) & A = A + F_P(\varepsilon) \\
 & F_P(u) \arrow[u] \arrow[ur, "(a_1, \dots, a_n)^{\#}"'] &
\end{tikzcd}$$

□

4.13. Definition. Let T be an S-sorted algebraic theory. Define $U : \mathbf{Alg}_T \to \mathbf{Set}^S$ to be the functor such that for each $\mathscr{A} \in |\mathbf{Alg}_T|$, $(U(\mathscr{A}))(s) = \mathscr{A}(s)$, and for each T-homomorphism, $h : \mathscr{A} \to \mathscr{B}$, $(U(h))_s = h_s$. We can show that U has a left adjoint F taking each S-ary set X to $F(X)$ the T-algebra freely generated by X. We will omit the proof, but note that by Proposition 4.11, F can be chosen so that for $w \in S^*$, $F(w) = F(\rho(w)) = T(w,\_)$.

4.14. Proposition. $\mathbf{Poly}_{T(\varepsilon,\_)} \cong T$.

Proof. Let T be an S-sorted theory, and let $v = v_1 \ldots v_n$, $w = w_1 \ldots w_p \in S^*$. Then

$$\begin{aligned}
\mathbf{Poly}_{T(\varepsilon,_)}(v, w) &= \mathbf{Set}^S(\rho(w), U(T(\varepsilon,_) + F(v))) && \text{definition of } \mathbf{Poly}\\
&= \mathbf{Set}^S(\rho(w), U(T(v,_))) && \text{Proposition 4.11 and Definition 4.13}\\
&= T(v, w). && \text{Definition 4.6}
\end{aligned}$$

It is then straightforward to show that the composition in $\mathbf{Poly}_{T(\varepsilon,\_)}$ and T are the same. □

5. Some simple programming languages as polynomial categories

Algebraic theories have been used for a number of purposes in theoretical computer science. The area of greatest use has probably been in the study of iteration, that is, in the mathematical treatment of programming languages involving iteration or recursion. See [6, 16, 7] for the historic origins of various approaches, see [14] for a motivational summary. Most of the work on iteration uses theories with additional structure, such as an order relation on the hom-sets, or an iteration operation with appropriate axioms. The theories of iteration do not make use of the algebras associated with the theories, but where both theories and their algebras are used in various approaches to data type specification (see [15, 3, 4]). In this paper we restrict our attention to some simple examples drawn from the theory of iteration. In particular, we give a treatment of flowcharts, followed by a briefer treatment of monadic recursion schemes. Both examples make use of an algebraic theory $\mathbf{Flow}_X$, which can be presented without introducing the additional structure developed in the above references.

5.1. Notation. A presentation $P = \langle S, \Sigma, E\rangle$ is said to be *1-sorted* if S has just one element. It is convenient then to identify S^* with $\omega = \{0, 1, 2, \ldots\}$, the set of natural numbers, and to index Σ by ω rather than by $S^* \times S$, so $\Sigma = \langle \Sigma_n \mid n \in \omega\rangle$.

Similarly, a *1-sorted algebraic theory* is an S-sorted algebraic theory where S has just one element. In general we will use $\omega \times \omega$ to index the hom sets of a 1-sorted algebraic theory.

5.2. Definition. Let Σ be the 1-sorted signature with $\Sigma_0 = \{\perp\}$, and with $\Sigma_n = \emptyset$ for $n \neq 0$. Let A be the initial Σ-algebra, so A has carrier $\{\perp\}$, and no operations other than the constant $\perp$. Now take a non-empty set X, and let $\mathcal{S} = \{X \times \{n\} \mid n \in \omega\} \subseteq |\mathbf{Set}|$.). Consider the polynomial category $\mathbf{Poly}_{A,\mathcal{S}}$. By Definition 3.11, a morphism in $\mathbf{Poly}_{A,\mathcal{S}}(X \times [n], X \times [p])$ is a mapping $X \times [p] \to U_\Sigma(A + F_\Sigma(X \times [n]))$. But $F_\Sigma(X \times [n])$ is just the Σ-algebra with carrier $U_\Sigma(F_\Sigma(X \times [n])) = \{\perp\} \cup (X \times [n])$.

and, since A is the initial Σ-algebra, $A + F_\Sigma(X \times [n]) = F_\Sigma(X \times [n])$ so $\mathbf{Poly}_{A,\mathcal{I}}(X \times [n], X \times [p])$ can be identified with the set of all mappings $X \times [p] \to \{\perp\} \cup (X \times [p])$. Given a morphism f, we shall write $\bar{f}$ for the corresponding mapping. Next, given $f \in \mathbf{Poly}_{A,\mathcal{I}}(X \times [n], X \times [p])$ and $g \in \mathbf{Poly}_{A,\mathcal{I}}(X \times [p], X \times [q])$, their composite is given by the mapping $\bar{h}: X \times [q] \to \{\perp\} \cup (X \times [n])$ such that

$$\bar{h}(\langle x, i\rangle) = \begin{cases} \bar{f}(\bar{g}(\langle x, i\rangle)) & \text{if } \bar{g}(\langle x, i\rangle) \neq \perp, \\ \perp & \text{if } \bar{g}(\langle x, i\rangle) = \perp. \end{cases}$$

Thus these mappings $\bar{f}: X \times [n] \to \{\perp\} \cup (X \times [p])$ are pointed functions, i.e. they preserve the point $\perp$, and, as such, can also be identified with the partial functions from $X \times [n] \rightharpoonup X \times [p]$.

Define $\mathbf{Flow}_X$ to be the category with $|\mathbf{Flow}_X| = \omega$, with $\mathbf{Flow}_X(n, p) = \mathbf{Poly}_{A,\mathcal{I}}(X \times [n], X \times [p])$, and with composition as in $\mathbf{Poly}_{A,\mathcal{I}}$.

5.3. Fact. *Given the category* $\mathbf{Flow}_X$ *as above, then for each* $n \in \omega$ *and each* $i \in [n]$, *define* $x_i^n \in \mathbf{Flow}_X(n, 1)$ *to be the morphism given by the mapping*

$$\bar{x}_i^n : X \times [1] \to X \times [n], \qquad \langle x, 1\rangle \mapsto \langle x, i\rangle$$

and given p *morphisms* $f_i \in \mathbf{Flow}(n, 1)$, $i = 1, \ldots, p$, *where* f_i *is given by a mapping* $\bar{f}_i : X \times [1] \to X \times [n]$, *define* $\langle f_1, \ldots, f_p\rangle : n \to p$ *to be the morphism in* $\mathbf{Flow}_X(n, p)$ *given by the mapping*

$$h : X \times [p] \to \{\perp\} \cup (X \times [n]), \qquad \langle x, i\rangle \mapsto \bar{f}_i(\langle x, 1\rangle).$$

Then $\mathbf{Flow}_X$ *equipped with these additional operations is a* 1*-sorted algebraic theory.*

It is convenient to generalize the tupling operation as follows: given $f : n \to p$ and $g : n \to q$ in $\mathbf{Flow}_X$ then the result, $h = \langle f, g\rangle : n \to p \quad q$, of tupling them together, is given by $\bar{h}$, where

$$\bar{h}(\langle x, i\rangle) = \begin{cases} \bar{f}(\langle x, i\rangle) & \text{if } i \leq p, \\ \bar{g}(\langle x, i - p\rangle) & \text{if } i > p. \end{cases}$$

There is an easy way to introduce an order structure on the morphisms of $\mathbf{Flow}_X$, and this ordering cooperates nicely with the operations of the theory.

5.4. Fact. *Let* X *be any non-empty set, and for each* $n, p \in \omega$ *let* $\leq_{n,p}$ *be the ordering on* $\mathbf{Flow}_X(p, n)$ *such that for* $\alpha_i \in \mathbf{Flow}_X(p, n)$, $i = 1, 2$, *with* α_i *given by* $\bar{\alpha}_i : X \times [n] \to \{\perp\} \cup (X \times [p])$, *then* $\alpha_1 \leq \alpha_2$ *iff for all* $\langle x, j\rangle \in X \times [n]$, $\bar{\alpha}_1(\langle x, j\rangle) \neq \perp$ *implies* $\bar{\alpha}_2(\langle x, j\rangle) = \bar{\alpha}_1(\langle x, j\rangle)$. *Then composition and tupling are monotonic with respect to* $\leq$, *that is,*

(1) *if* α_1, $\beta_1 : \mathbf{Flow}_X(n, p)$ *and* α_2, $\beta_2 \in \mathbf{Flow}_X(p, q)$ *with* $\alpha_i \leq \beta_i$, $i = 1, 2$, *then* $\alpha_2 \bullet \alpha_1 \leq \beta_2 \bullet \beta_1$;

(2) *if* α_1, $\alpha_2 \in \mathbf{Flow}_X(n, p)$ *and* β_1, $\beta_2 \in \mathbf{Flow}_X(n, q)$ *with* $\alpha_i \leq \beta_i$, $i = 1, 2$, *then* $\langle \alpha_1, \alpha_2\rangle \leq \langle \beta_1, \beta_2\rangle$.

Furthermore, if $\langle \alpha_i : n \to p \mid i \in \omega \rangle$ *is an* ω*-chain with respect to* $\leqslant$, *i.e. if* $\alpha_i \leqslant \alpha_{i+1}$ *for all* $i \in \omega$, *then* $\langle \alpha_i : n \to p \mid i \in \omega \rangle$ *has a least upper bound* $\bigsqcup \langle \alpha_i : n \to p \mid i \in \omega \rangle$. *Lastly, composition is left continuous, i.e. for any* ω*-chain* $\langle \alpha_i : n \to p \mid i \in \omega \rangle$ *and any morphism* $\beta : p \to q$,

$$\bigsqcup \langle \beta \bullet \alpha_i : n \to q \mid i \in \omega \rangle = \beta \bullet \bigsqcup \langle \alpha_i : n \to p \mid i \in \omega \rangle.$$

and tupling is continuous, i.e. for any two ω*-chains* $\langle \alpha_i : n \to p \mid i \in \omega \rangle$ *and* $\langle \beta_i : n \to q \mid i \in \omega \rangle$, *we have*

$$\bigsqcup \langle \langle \alpha_i, \beta_i \rangle \mid i \in \omega \rangle = \langle \bigsqcup \langle \alpha_i \mid i \in \omega \rangle, \bigsqcup \langle \beta_i \mid i \in \omega \rangle \rangle.$$

We can interpret $\mathbf{Flow}_X$ in a natural way as a *theory of flowcharts.* Think of X as "the set of states". Then each element α of $\mathbf{Flow}_X(n, p)$ can be interpreted as a black-box with p input channels and n output channels (note the reversal of n and p) such that, when we enter the ith input channel in state x, and $\bar{\alpha}(\langle x, i \rangle) = \langle y, j \rangle \in X \times [n]$, then we exit the block in state y on the jth output channel, while if $\bar{\alpha}(\langle x, i \rangle) = \perp$ then we say that the output is undefined. We can picture this as shown in Fig. 3(a). In particular, we can view a morphism in $\mathbf{Flow}_X(1, 1)$ as a one-input,

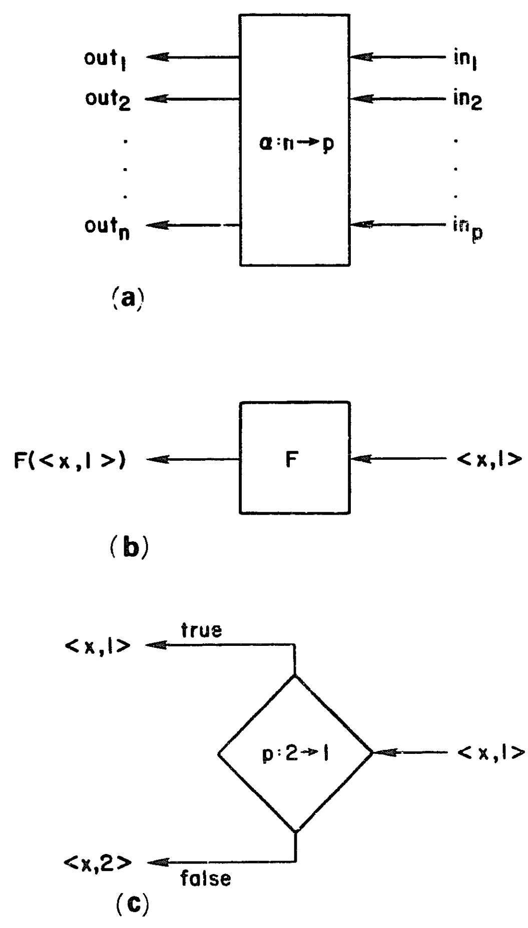

Fig. 3.

one-output black box (Fig. 3(b)) and, more important, and more interesting, given a predicate P on X, we can construct a morphism $p \in \mathbf{Flow}(2, 1)$ corresponding to the conditional-block for P, by taking $p = \langle 2, \bar{p}, 1\rangle$ where $\bar{p}: X \times [1] \to X \times [2]$, such that (cf. Fig. 3(c))

$$\bar{p}(\langle x, 1\rangle) = \begin{cases} \langle x, 1\rangle & \text{if } P(x), \\ \langle x, 2\rangle & \text{if } \neg P(x). \end{cases}$$

Given a collection of morphisms $f_i: 1 \to 1$ and $p_j: 2 \to 1$ from $\mathbf{Flow}_X$, then the subcategory of $\mathbf{Flow}_X$ generated by this collection using composition and tupling corresponds to flowcharts that can be built up using the corresponding boxes. How this works is easy to see by taking the following specific choice for X, and for the generating morphisms.

Let $\mathrm{Id} = \{A, B, C, \ldots\}$ be a set (of *identifiers*), take $X = \mathbf{Set}(\mathrm{Id}, \omega)$ where ω denotes the set of natural numbers, $\omega = \{0, 1, 2, \ldots\}$. That is, the set of states we will use will be the set of all mappings from the set of identifiers to the set of natural numbers. We now want to produce a simple programming language for defining state transformations in terms of flowcharts using arithmetic expressions, simultaneous assignments, and predicates "ι equals 0" for $\iota \in \mathrm{Id}$. To do this formally we define Σ to be the 1-sorted signature with

$$\Sigma_0 = \omega, \qquad \Sigma_1 = \{\mathrm{succ}, \mathrm{pred}\}, \qquad \Sigma_2 = \{+, *, -\},$$

and let **Arith** be the Σ-algebra with the set ω of natural numbers as carrier and with $\mathrm{succ}_{\mathbf{Arith}}$, $\mathrm{pred}_{\mathbf{Arith}}$, $+_{\mathbf{Arith}}$, $*_{\mathbf{Arith}}$, and $-_{\mathbf{Arith}}$ being respectively, the arithmetic operations successor, predecessor, addition, multiplication and subtraction. A simultaneous assignment can be defined as a mapping

$$\alpha: \mathrm{Id} \to U(\mathbf{Arith} + \mathrm{Fr}(\mathrm{Id})).$$

Any such α induces a state-to-state mapping $\bar{\alpha}: X \to X$ in the same way as in Example 3.12, and $\bar{\alpha}$ can, in turn, be viewed as morphism in $\mathbf{Flow}_X(1, 1)$ because X and $X \times [1]$ are isomorphic sets. Furthermore, the conditional for the predicate "ι equals 0", corresponds to the mapping

$$\begin{aligned} &\mathrm{If}_\iota : X \times [1] \to X \times [2], \\ &\quad \langle x, 1\rangle \mapsto \langle x, 1\rangle \quad \text{if } x(\iota) = 0, \\ &\quad \langle x, 1\rangle \mapsto \langle x, 2\rangle \quad \text{if } x(\iota) \neq 0. \end{aligned}$$

Then a flowchart of the form as shown in Fig. 4(a) can be represented by a morphism in $\mathbf{Flow}_X$ by suitably "cutting" the loops, and indicating the connections. That is, we can cut up the flowchart as depicted in Fig. 4(b) which corresponds to the morphism γ in $\mathbf{Flow}_X(4, 3)$ given by the mapping

$$\begin{aligned} &\bar{\gamma}: X \times [3] \to X \times [4], \\ &\quad \langle x, 1\rangle \mapsto \langle (A := 1)(x), 2\rangle, \end{aligned}$$

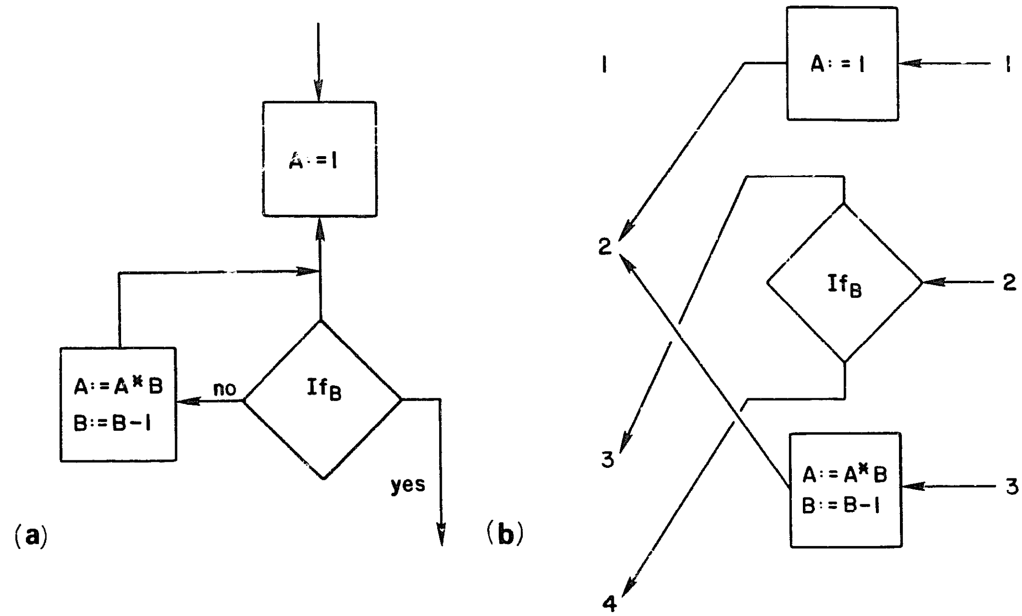

Fig. 4.

$$\langle x, 2\rangle \mapsto \begin{cases} \langle x, 3\rangle & \text{if } B(x) = 0, \\ \langle x, 4\rangle & \text{if } B(x) \neq 0, \end{cases}$$

$$\langle x, 3\rangle \mapsto \langle (A, B := A * B, B - 1)(x), 2\rangle,$$

where

$$(A := 1): X \to X, \qquad x \mapsto x'$$

where

$$x': A \mapsto 1, \qquad x': \iota \mapsto x(\iota), \quad \text{for all } \iota \in \mathrm{Id}, \iota \neq A$$

and similarly,

$$(A, B := A * B, B - 1): X \to X, \qquad x \mapsto x'$$

where

$$x': A \mapsto x(A) * x(B), \qquad x': B \mapsto x(B) - 1,$$

$$x': \iota \mapsto x(\iota), \quad \text{for all } \iota \in \mathrm{Id}, \iota \neq A.$$

Using the above approach we can represent a given, cut flowchart as a morphism α in $\mathbf{Flow}_X$. From this morphism α we can construct another morphism $\alpha^\dagger$ in $\mathbf{Flow}_X$ that represents the behavior, or semantics, of the flowchart. The construction is a particular case of the least fix point construction used throughout programming language semantics. For a more general treatment see the above references.

5.5. Theorem. *Let X be any non-empty set, and let $\leq_{p,n}$ be the ordering on $\mathbf{Flow}_X(p, n)$. Then for each $\alpha :\in \mathbf{Flow}_X(n+p, n)$ there exists a least $\alpha^\dagger \in \mathbf{Flow}_X(p, n)$ with respect to $\leq_{p,n}$ such that*

$$\alpha \bullet \langle \alpha^\dagger, 1_p \rangle = \alpha^\dagger.$$

Proof. We will take $\alpha^\dagger$ to be the least upper bound, with respect to $\leqslant$, of the ω-indexed family, $\langle\alpha^{(k)}: n\to p \mid k\in\omega\rangle$, of elements of $\mathbf{Flow}_X(n,p)$, where $\bar{\alpha}^{(0)}: Z\times[p]\to\{\perp\}\cup(X\times[n])$ takes everything to $\perp$, and where for all $k\geqslant 0$,

$$\alpha^{(k+1)}=\alpha\bullet\langle\alpha^{(k)},1_p\rangle.$$

To prove this works we will first show that $\langle\alpha^{(k)}\mid k\in\omega\rangle$ is an ω-chain, then taking $\alpha^\dagger$ to be the upper bound of this ω-chain we show that it is the least element $\xi\in\mathbf{Flow}_X(p,n)$ such that $\alpha\bullet\langle\xi,1_p\rangle=\xi$.

That $\langle\alpha^{(k)}\mid k\in\omega\rangle$ is an ω-chain is proved by induction on k. That $\alpha^{(0)}\leqslant\alpha^{(1)}$ follows immediately from the definitions of $\alpha^{(0)}$ and $\leqslant$, and assuming $\alpha^{(k)}\leqslant\alpha^{(k+1)}$ we obtain

$$\alpha^{(k+1)}=\alpha\bullet\langle\alpha^{(k)},1_p\rangle\leqslant\alpha\bullet\langle\alpha^{(k+1)},1_p\rangle=\alpha^{(k+2)}$$

since composition and tupling in $\mathbf{Flow}_X$ are monotonic.

Since $\langle\alpha^{(k)}\mid k\in\omega\rangle$ is an ω-chain we know, by Fact 5.4, that it has a least upper bound $\alpha^\dagger$. To see that $\alpha\bullet\langle\alpha^\dagger,1_p\rangle=\alpha^\dagger$ we observe that

$$\begin{aligned}
\alpha^\dagger &= \bigsqcup\langle\alpha^{(k)}\mid k\in\omega\rangle && \text{definition of } \alpha^\dagger\\
&= \bigsqcup\langle\alpha\bullet\langle\alpha^{(k)},1_p\rangle\mid k\in\omega\rangle && \text{definition of } \alpha^{(k)}\\
&= \alpha\bullet\bigsqcup\langle\langle\alpha^{(k)},1_p\rangle\mid k\in\omega\rangle && \text{continuity of } \bullet\\
&= \alpha\bullet\langle\bigsqcup\langle\alpha^{(k)}\mid k\in\omega\rangle,1_p\rangle && \text{continuity of } \langle\ldots\rangle\\
&= \alpha\bullet\langle\alpha^\dagger,1_p\rangle && \text{definition of } \alpha^\dagger.
\end{aligned}$$

Thus $\alpha^\dagger$ is a fix point, to see that it is the least fix point let ξ be any other morphism such that $\alpha\bullet(\xi,1_p)=\xi$. Clearly $\alpha^{(0)}\leqslant\xi$ and $\alpha^{(k)}\leqslant\xi$ implies

$$\alpha^{(k+1)}=\alpha\bullet\langle\alpha^{(k)},1_p\rangle\leqslant\alpha\bullet\langle\xi,1_p\rangle=\xi$$

by the monotonicity of $\bullet$ and $\langle\ldots\rangle$, and thus ξ is an upper bound for $\langle\alpha^{(k)}\mid k\in\omega\rangle$ and thus $\alpha^\dagger\leqslant\xi$. $\square$

Applying this to the above example where $\alpha: 4\to 3$, we obtain $\alpha^\dagger: 1\to 3$, where for any state $x:\mathrm{Id}\to\omega$, $\alpha^\dagger(\langle x,i\rangle)$ will be the result of entering the flowchart on channel $i\in[3]$ in state x. So, in particular, $\alpha^\dagger(\langle x,1\rangle)$ should be the result of entering the flowchart "at the top" in state x. Working through the definition will show that the result, x', is the state such that

$$x'(A)=(x(B))!,\qquad x'(B)=0,$$

$$x'(\iota)=x(\iota),\quad\text{for all } \iota\in\mathrm{Id},\ \iota\neq A,B.$$

We constructed $\mathbf{Flow}_X$ by constructing a category of polynomials over a very simple algebra. Since $\mathbf{Flow}_X$ is an algebraic theory it is also an algebra and we can, of course, construct new categories and algebraic theories of polynomials over $\mathbf{Flow}_X$. It is natural to ask if this kind of construction can be used to construct higher level languages over $\mathbf{Flow}_X$. The answer is yes, and we will look briefly at one example, the construction of a language of monadic recursion schemes. The idea is to construct a new one-sorted theory $\mathbf{Mon}_X$ from $\mathbf{Flow}_X$ by adjoining "function variables", f_i, to $\mathbf{Flow}_X(1,1)$. More precisely, take $\mathcal{T} = \langle \mathcal{T}_n = \{f_1^n, \ldots, f_n^n\} \mid n \in \omega \rangle$, then a morphism in $\alpha \in \mathbf{Mon}_X(n,p)$ will be given by a mapping

$$\bar{\alpha} : \mathcal{T}_p \to (\mathbf{Flow}_X + \mathcal{F}(\mathcal{T}_n))(1,1)$$

where $\mathcal{F}(\mathcal{T}_n)$ is the free 1-sorted algebraic theory generated by the signature Σ with $\Sigma_1 = \mathcal{T}_n$, and with $\Sigma_m = \emptyset$ for $m \neq 1$.

For example, using the special choice of X given earlier, we have a morphism α in $\mathbf{Mon}_X(3,3)$ such that

$$\bar{\alpha}(f_1^3) = (A := 1) \bullet f_2^3,$$

$$\bar{\alpha}(f_2^3) = \mathrm{If}_B(1, (A, B := A * B, B-1) \bullet f_2^3),$$

$$\bar{\alpha}(f_3^3) = f_1^3 \bullet (B := A) \bullet f_1^3.$$

Composition and finite products in $\mathbf{Mon}_X$ are defined as in Section 3; just take $\mathcal{T}_n + \mathcal{T}_p = \mathcal{T}_{n+p}$, with appropriate injections, to obtain the coproduct needed for Proposition 3.14. This makes $\mathbf{Mon}_X$ into an algebraic theory, and the order structure on $\mathbf{Flow}_X$ induces an order structure on $\mathbf{Mon}_X$ satisfying the properties given in Fact 5.4, and so we can take fix points in the manner of Theorem 5.5. The fix point $\alpha^\dagger : 0 \to 3$ for the above $\alpha : 3 \to 3$ will be such that for an $x : \mathrm{Id} \to \omega$,

$$((\bar{\alpha}^\dagger(f_3^3))(x))(A) = ((x(B))!)!$$

the factorial of the factorial of the value of (B) in state x.

Of course, we can also form polynomials over $\mathbf{Mon}_X$ to obtain additional theories. Doing this in the manner of Section 4 will yield an $\omega \times \omega$-sorted theory in which higher-level operators such as WHILE-DO appear directly as morphisms of the theory.

References

[1] M. Barr and C. Wells, *Category Theory for Computing Science* (Prentice-Hall, Englewood Cliffs, 1990).

[2] S.L. Bloom, J.W. Thatcher, E.G. Wagner and J.B. Wright, Recursion and iteration in continuous theories: The M-construction, *J. Comput. System Sci.* **27** (1983) 148–164.

[3] S.L. Bloom and E.G. Wagner, Many-sorted theories and their algebras with some applications to data types, in: M. Nivat and J.C. Reynolds, eds., *Algebraic Methods in Semantics* (Cambridge University Press, Cambridge, 1985) 133–168.

[4] R.M. Burstall and J.A. Goguen, Putting theories together to make specifications, in: *Proc. IJCAI-77* Cambridge, MA (1977) 292–332.

[5] P.M. Cohn, *Universal Algebra* (Harper and Row, New York, 1965).

[6] C.C. Elgot, Monadic computation and iterative algebraic theories, in: H.E. Rose and J.S. Shepherdson, eds., *Proc. Logic Colloquium '73, Studies in Logic* (North-Holland, Amsterdam, 1975) 175–230.

[7] Z. Esik, Identities in iterative and rational algebraic theories, *Comput. Linguistics Comput. Languages* **14** (1980) 183-207.
[8] G. Grätzer, *Universal Algebra* (Springer-Verlag, Berlin, 2nd ed., 1979).
[9] H. Herrlich and G.E. Strecker, *Category Theory* (Allyn and Bacon, Boston, 1973).
[10] W. Lawvere, Functorial semantics of algebraic theories, *Proc. Nat. Acad. Sci.* **21** (1963) 1-23.
[11] R.N. McKenzie, G.F. McNulty and W.F. Taylor, *Algebras, Lattices, Varieties, Vol. 1* (Wadsworth and Brooks-Cole, Monterey, 1987).
[12] S. MacLane, *Categories for the Working Mathematician* (Springer-Verlag, New York, 1971).
[13] E.G. Manes, *Algebraic Theories* (Academic Press, New York, 1976).
[14] E.G. Wagner, S.L. Bloom and J.W. Thatcher, Why algebraic theories, in: M. Nivet and J.C. Reynolds, eds., *Algebraic Methods in Semantics* (Cambridge University Press, Cambridge, 1985) 607-634.
[15] E.G. Wagner and H. Ehrig, Canonical constraints for parameterized data types, *Theoret. Comput. Sci.* **50** (1987) 323-349.
[16] J.B. Wright, J.W. Thatcher, E.G. Wagner and J.A. Goguen, 'Rational algebraic theories and fixed-point solutions, in: *Proc. 17th IEEE Symp. on Foundations of Computing*, Houston, Texas (1976) 147-158.