Congruences and Quotients in Categories of Algebras

Nico Verwer

RUU-CS-92-23
June 1992

Utrecht University
Department of Computer Science
Padualaan 14, P.O. Box 80.089,
3508 TB Utrecht, The Netherlands,
Tel. : ... + 31 - 30 - 531454

Congruences and Quotients in Categories of Algebras

Nico Verwer

Technical Report RUU-CS-92-23
June 1992

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

ISSN: 0924-3275

Congruences and Quotients in Categories of Algebras

Nico Verwer

Department of Computer Science, Utrecht University,
P.O. box 80.089, 3508 TB Utrecht, the Netherlands.
Email: nico@cs.ruu.nl.

Abstract

Equational specifications can be implemented by quotient algebras. A category theoretical description of algebraic specification should include quotient algebras in such a way that it can serve as a guide in building algebraic specification systems. In this paper we give a new construction which does exactly this, and give an example of how it could be used. This construction is simple, but still applicable in practical situations.

1 Introduction

The category theoretical description of abstract data types is elegant, because it has a high level of abstraction, and gives a concise and precise definition of the notions that are involved, and the way they are related. But it is more than a nice exercise in theoretical computer science; a thorough category theoretical description can be used to implement a system for compiling and manipulating algebraic specifications. Such a system tends to be complex, and a clear description can help tremendously in building it, by defining interfaces for the large number of modules involved. Without a clear view of the different aspects of the system, construction is cumbersome, ad hoc, and of limited applicability. The machine, or programming language in which the system is implemented is viewed as a *base category*, upon which the complicated constructions are made. Along the lines of 'computational category theory' [7], a system for algebraic specification and implementation of data types is being built at the moment.

The purpose of this paper is to propose a category theoretical description of *quotient algebras*, as well as to give constructions for it. Quotient algebras are important for the specification and implementation of abstract data types, because they are models of equational specifications. In a quotient algebra the values are equivalence classes with respect to a *congruence relation*. The important thing about a congruence relation is that it preserves the operations of the algebra, a property which is usually called *compatibility*.

Several ways of describing quotients in a category theoretical framework have been devised. In the sketches approach of Barr and Wells [1], congruences are formed by a construction that is outside the base category. Since we would like to do all programming within this category, this approach is not applicable in our case.

In the category **Set**, a construction for quotient algebras is given by Ehrig and Mahr [3]. This construction relies on the fact that epimorphisms split (i.e. we can use an axiom of choice in our programs), something which we cannot always assume. We discuss this in section 5.

Fokkinga explores algebras in a more general framework in [4]. In his chapter on 'laws', he restricts relations to the subcategory $Congr(\phi)$ of congruence relations on the algebra ϕ, but again this is a restriction imposed from outside. Also it is not clear whether there is a category theoretical way of deciding what relations are congruences.

The approach developed here is similar to that of Diaconescu in [2]. It may seem less powerful than those just mentioned, but this is only because we make no assumptions which cannot be satisfied easily: It shows exactly where the hard work in building systems for algebraic specification takes place. This is illustrated in section 5. Here we are interested in the motivation behind the constructions, and their applicability, whereas Diaconescu concentrates on soundness and completeness properties.

2 Categories and algebras

It is assumed that the reader has an elementary knowledge of category theory on the level of, for instance, the introductions of Rydeheard & Burstall [7], or Pierce [5]. The reader should also know something about the theory of algebraic specification and implementation of data types. There is a great amount of literature on this subject; the introductory chapters of Ehrig & Mahr [3] and Wirsing [8] provide enough background to understand this paper.

The notation used here is rather non-standard for category theorists. Some of it is borrowed from Fokkinga's thesis [4], and some of it agrees with what is common in algebraic specification formalisms.

Objects in a category $\mathcal{C}$ are indicated with capitals A, B, …or greek letters ϕ, ψ, …if they are algebras. Arrows are indicated by f, g, h, …or ϕ, ψ, …. Composition of arrows is written as $\circ$, and is in anti-diagramatic order. Functors are indicated by whatever symbol is appropriate. Functor application is done by superscripting, like in A^F or f^F. Functor composition is in diagramatic order, and written as ; or just by juxtaposition. When τ is a natural transformation from F to G, this is written as $\tau : F \dot{\to} G$. For every object $A \in \mathcal{C}$, there is a *constant functor*

$\underline{A} : \mathcal{D} \longrightarrow \mathcal{C}$, defined by

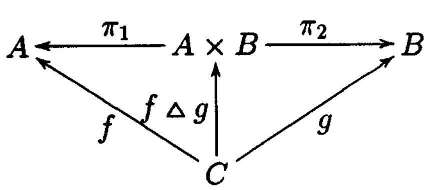
$$X^{\underline{A}} = A \;, \quad f^{\underline{A}} = \mathrm{id}_A.$$

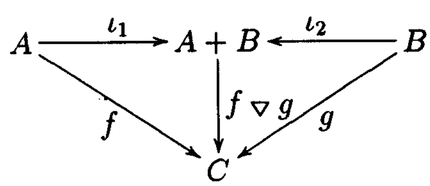
Here $\mathcal{D}$ may be any category, and $F ; \underline{X} = \underline{X}$. The *lifting* of a bifunctor $\oplus$ is defined as

$$\begin{array}{l} F \hat{\oplus} G : \mathcal{C} \rightarrow \mathcal{E} \\ A^{F \hat{\oplus} G} = A^F \oplus A^G \;, \quad f^{F \hat{\oplus} G} = f^F \oplus f^G \end{array}$$

for $F, G : \mathcal{C} \longrightarrow \mathcal{D}$ and $\oplus : \mathcal{D} \times \mathcal{D} \longrightarrow \mathcal{E}$.

We use the notation $\mathbf{1}$ for the terminal object in a category, and $!_A : A \rightarrow \mathbf{1}$ for the unique arrow from the object A to it. The initial object, and the unique arrow from it to A are denoted by $¡_A : \mathbf{0} \rightarrow A$. The notation for product and coproduct is as in the diagram below: [1]





A category $\mathcal{C}$ is called *extensional* if for all objects A

$$\forall a : \mathbf{1} \rightarrow A : f \circ a = g \circ a \;\Rightarrow\; f = g.$$

Sometimes this is expressed as "$\mathbf{1}$ is a *generator*."

For a cartesian closed category $\mathcal{C}$, the exponential is written $B \leftarrow A$. Currying is written as $^{\copyright}$, and application as @. The *name* of an arrow $f : A \rightarrow B$ is

$$`f' = (f \circ \pi_{2(\mathbf{1},A)})^{\copyright} : \mathbf{1} \rightarrow (B \leftarrow A).$$

From now on, $\mathcal{C}$ is the *base category*, i.e. the model of the machine on which programs are executed. We assume that it is cartesian closed, and that it has finite coproducts. It should also be extensional.

A *signature* is represented by an endofunctor $\Sigma : \mathcal{C} \rightarrow \mathcal{C}$. A Σ-algebra is an arrow $\phi : A^{\Sigma} \rightarrow A$ in $\mathcal{C}$. The object A is the carrier of ϕ, and we shall write it as $\phi^{||}$. A homomorphism h from the algebra ϕ to the algebra ψ is determined by an arrow $h^{||} : \phi^{||} \rightarrow \psi^{||}$ in $\mathcal{C}$, such that

$$h^{||} \circ \phi = \psi \circ h^{||\Sigma}.$$

The category $\mathcal{C}|\Sigma$ has Σ-algebras as objects, and homomorphisms as arrows. Identity and composition are taken from $\mathcal{C}$. The carrier is a functor $|| : \mathcal{C}|\Sigma \rightarrow \mathcal{C}$.

[1] The diagrams in this paper were drawn with the XY-pic package [6].

The natural transformation

$$\boldsymbol{i} : \| \,;\Sigma \dashrightarrow \|$$

is defined as

$$\boldsymbol{i}_\phi = \phi$$

where the subscript ϕ (in which $\boldsymbol{i}$ is instantiated) is the algebra, and the other ϕ is the arrow in $\mathcal{C}$. A signature $\Sigma : \mathcal{C} \longrightarrow \mathcal{C}$ gives rise to a functor

$$\begin{aligned} &\boldsymbol{\Sigma} : \mathcal{C}|\Sigma \longrightarrow \mathcal{C}|\Sigma \\ &\phi^{\boldsymbol{\Sigma}} = \phi^{\Sigma} : \phi^{\|\Sigma\Sigma} \longrightarrow \phi^{\|\Sigma}. \\ &h^{\boldsymbol{\Sigma}\|} = h^{\|\Sigma} \end{aligned}$$

This means that

$$\phi^{\boldsymbol{\Sigma}\|} = \phi^{\|\Sigma}.$$

From now on we shall write Σ instead of $\boldsymbol{\Sigma}$.

The initial Σ-algebra in $\mathcal{C}|\Sigma$ is $\mu(\Sigma) : \mu(\Sigma)^{\|\Sigma} \longrightarrow \mu(\Sigma)^{\|}$. It does not always exist, but its existence can be proved if $\mathcal{C}$ is ω-cocomplete and Σ is ω-cocontinuous. There is a unique homomorphism $\text{!`}_\phi : \mu(\Sigma) \rightarrow \phi$ for every Σ-algebra ϕ. In the notation of [4],

$$(\!|\phi|\!) = \text{!`}_\phi{}^{\|} : \mu(\Sigma)^{\|} \rightarrow \phi^{\|}$$

which is called the catamorphism of ϕ.

3 Equations and variables

In most algebraic specification languages, the simplest form of equation is a pair of terms. When the base category is **Set**, terms are the elements of the term algebra over the signature Σ. This is the free initial algebra $\mu(\Sigma)$; its carrier object is the set of all well-formed expressions with operators from Σ. Thus, category theoretically, a (ground) term corresponds to an arrow

$$t : \mathbf{1} \rightarrow \mu(\Sigma)^{\|}.$$

Definition 1 Let $t\,, t' : \mathbf{1} \rightarrow \mu(\Sigma)^{\|}$. The property that the equation $t = t'$ *holds* in the algebra ϕ (ϕ *satisfies* $t = t'$) is defined as

$$\phi \models t \sim t' \;\Leftrightarrow\; (\!|\phi|\!) \circ t = (\!|\phi|\!) \circ t'.$$

□

We used the symbol '$\sim$' instead of '$=$' in this definition, because the terms t and t' themselves are not equal; their interpretations in the algebra ϕ are equal.

The situation becomes more complicated when we allow *variables* to occur in the equations. Set-theoretically, the variables are just elements of some set (in some specification language this set is enumerated in the **vars** section of a specification). An obvious generalization is to consider a 'variable-object', which can be any object in the base category. Individual variables are constant arrows $x : \mathbf{1} \rightarrow X$ to this object. The signature of Σ-algebras with variables in X is

$$\Sigma \hat{+} \underline{X} : \mathcal{C} \rightarrow \mathcal{C}.$$

The variables are considered as new constants in the signature. The algebra of Σ-terms with variables in X is the initial $\Sigma \hat{+} \underline{X}$-algebra

$$\mu(\Sigma \hat{+} \underline{X}) : (\mu(\Sigma \hat{+} \underline{X}))^{||\Sigma} + X \rightarrow (\mu(\Sigma \hat{+} \underline{X}))^{||}$$

The interesting thing about variables is that we can assign values to them. This is done by something called a valuation, assignment, substitution, or environment. Here we shall use the word *assignment* for a natural transformation

$$\alpha : \underline{X} \dot{\rightarrow} ||$$

(note that $\underline{X} : \mathcal{C}|\Sigma \longrightarrow \mathcal{C}$). This may be instantiated in an algebra ϕ to get $\alpha_\phi : X \rightarrow \phi^{||}$. Every arrow $f : X \rightarrow \mu(\Sigma)^{||}$ can be extended to an assignment by defining $\alpha_\phi = (\!|\phi|\!) \circ f$.

Sometimes we would like to have a variable object X that is not fixed, but dependent on the algebra to which the variables are added. In this case we do not have a variable object X, but a variable functor $F : \mathcal{C} \longrightarrow \mathcal{C}$. The extended signature is $\Sigma \hat{+} F$, and an assignment is a natural transformation $\alpha : || \, ; F \dot{\rightarrow} ||$. When the variables are fixed, the functor F is $\underline{X}$.

We shall now introduce functors to make $\Sigma \hat{+} F$-algebras from Σ-algebras, and vice versa.

Definition 2 (variable addition) Let ϕ and ψ be Σ-algebras, $h : \phi \rightarrow \psi$ a Σ-homomorphism, and F a functor, $F : \mathcal{C} \longrightarrow \mathcal{C}$. For every assignment $\alpha : || \, ; F \dot{\rightarrow} ||$ there is a functor

$$[\alpha] : \mathcal{C}|\Sigma \longrightarrow \mathcal{C}|(\Sigma \hat{+} F)$$

defined by

$$\phi^{[\alpha]} = \phi \triangledown \alpha_\phi : \phi^{||\Sigma} + \phi^{||F} \rightarrow \phi^{||}$$
$$h^{[\alpha]||} = h^{||}$$

□

This defines a homomorphism $h^{[\alpha]}$, because

$$
\begin{aligned}
& h^{[\alpha]||} \circ \phi^{[\alpha]} = \psi^{[\alpha]} \circ (h^{[\alpha]||\Sigma} + h^{[\alpha]||F}) \\
\Leftrightarrow\ & (\text{definitions of } h^{[\alpha]} \text{ and } [\alpha]) \\
& h^{||} \circ (\phi \triangledown \alpha_\phi) = (\psi \triangledown \alpha_\psi) \circ (h^{||\Sigma} + h^{||F}) \\
\Leftrightarrow\ & \\
& (h^{||} \circ \phi) \triangledown (h^{||} \circ \alpha_\phi) = (\psi \circ h^{||\Sigma}) \triangledown (\alpha_\psi \circ h^{||F}) \\
\Leftrightarrow\ & \\
& h^{||} \circ \phi = \psi \circ h^{||\Sigma} \ \wedge\ h^{||} \circ \alpha_\phi = \alpha_\psi \circ h^{||F} \\
\Leftarrow\ & (h \text{ is a homomorphism, } \alpha \text{ a natural transformation}).
\end{aligned}
$$

Definition 3 (variable deletion) Let $\phi\,,\psi$ be $(\Sigma \hat{+} F)$-algebras, and $h : \phi \to \psi$. The functor

$$\backslash F : \mathcal{C}|(\Sigma \hat{+} F) \longrightarrow \mathcal{C}|\Sigma$$

is defined as

$$
\begin{aligned}
& \phi^{\backslash F} = \phi \circ \iota_1 : \phi^{||\Sigma} \longrightarrow \phi^{||} \\
& h^{\backslash F||} = h^{||}.
\end{aligned}
$$

□

It is easy to see that $h^{\backslash F}$ is a homomorphism:

$$
\begin{aligned}
& h^{\backslash F||} \circ \phi^{\backslash F} = \psi^{\backslash F} \circ h^{\backslash F||\Sigma} \\
\Leftrightarrow\ & (\text{definition of } \backslash F) \\
& h^{||} \circ \phi \circ \iota_1 = \psi \circ \iota_1 \circ h^{||\Sigma} \\
\Leftarrow\ & (\text{coproduct property}) \\
& h^{||} \circ \phi = \psi \circ (h^{||\Sigma} + h^{||F}) \\
\Leftarrow\ & (h \text{ is a homomorphism}).
\end{aligned}
$$

Both variable addition and deletion are natural with respect to the base category $\mathcal{C}$.

The following properties can be used to calculate with variable addition and deletion:

Proposition 1 For every assignment $\alpha : || \,;\, F \dot{\to} ||$

$$[\alpha] \,;\, \backslash F = \mathbf{I}.$$

The other way round, this holds for a very specific assignment only:

$$\backslash F \,;\, [\boldsymbol{\imath} \circ \iota_2] = \mathbf{I}.$$

□

Proof For algebras, all we have to do is to write out the definitions:

$$\phi^{[\alpha]\backslash F} =$$
$$(\phi \triangledown \alpha_\phi) \circ \iota_1 =$$
$$\phi$$

and

$$\psi^{\backslash F[\mathbf{i}\circ\iota_2]} =$$
$$(\psi \circ \iota_1) \triangledown (\psi \circ \iota_2) =$$
$$\psi$$

For homomorphisms the proposition is trivial.
□

There are also some interesting properties of initial algebras with variables:

Proposition 2 If ϕ is a Σ-algebra and $\alpha : \| ; F \dashrightarrow \|$, then

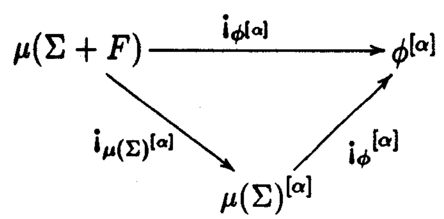
$$\mathbf{i}_\phi{}^{[\alpha]} \circ \mathbf{i}_{(\mu(\Sigma)^{[\alpha]})} = \mathbf{i}_{(\phi^{[\alpha]})} \quad \text{in } \mathcal{C}|(\Sigma \hat{+} F)$$
$$\mathbf{i}_\phi \circ \mathbf{i}_{(\mu(\Sigma)^{[\alpha]})}{}^{\backslash F} = \mathbf{i}_{(\phi^{[\alpha]})}{}^{\backslash F} \quad \text{in } \mathcal{C}|\Sigma.$$

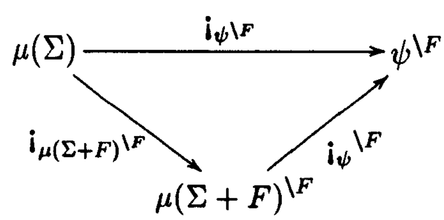
If ψ is a $(\Sigma \hat{+} F)$-algebra, then

$$\mathbf{i}_\psi{}^{\backslash F} \circ \mathbf{i}_{(\mu(\Sigma\hat{+}F)^{\backslash F})} = \mathbf{i}_{(\psi^{\backslash F})} \quad \text{in } \mathcal{C}|\Sigma$$
$$\mathbf{i}_\psi \circ \mathbf{i}_{(\mu(\Sigma\hat{+}F)^{\backslash F})}{}^{[\mathbf{i}\circ\iota_2]} = \mathbf{i}_{(\psi^{\backslash F})}{}^{[\mathbf{i}\circ\iota_2]} \quad \text{in } \mathcal{C}|(\Sigma \hat{+} F).$$

□

The proofs follow immediately from the initiality of $\mathbf{i}$ and proposition 1. In order to check this, the reader may find the following diagrams helpful.





Proposition 3 (free algebra) The carrier functor $\| : \mathcal{C}|\Sigma \to \mathcal{C}$ is a forgetful functor, which has a left adjoint $\mathcal{F}_\Sigma : \mathcal{C} \to \mathcal{C}|\Sigma$, which is defined as follows:

$$A^{\mathcal{F}_\Sigma} = \mu(\Sigma \hat{+} \underline{A})^{\backslash \underline{A}}, \; A \in \mathcal{C}_o$$
$$f^{\mathcal{F}_\Sigma} = \mathbf{i}_{(\mu(\Sigma\hat{+}\underline{B})\circ(\mathrm{id}+f))}{}^{\backslash \underline{A}}, \; f : A \to B$$

□

This is illustrated in the following diagram.

$$\begin{array}{ccc} \mathbf{1} & \xrightarrow{\quad t \quad} & \mu(\Sigma \hat{+} F)^{||} \\ {\scriptstyle `\alpha_\phi'}\big\downarrow & & \big\downarrow{\scriptstyle (\!|\phi^{[\alpha]}|\!)} \\ \phi^{||} \leftarrow \phi^{||F} & \xrightarrow[\sqrt{t_\phi}]{\qquad} & \phi^{||} \end{array}$$

The activation is unique in extensional categories.

Proposition 4 If $\mathbf{1}$ is a generator in $\mathcal{C}$ ($\mathcal{C}$ is extensional) then $\sqrt{t_\phi}$ is unique if it exists. Moreover, if it exists for all algebras, then it is a natural transformation

$$\sqrt{t} : || \dot{\leftarrow} (|| \, ; F) \dot{\rightarrow} ||.$$

□

The following proposition shows how activations can be constructed for polynomial signatures.

Proposition 5 (construction of activations) If Σ is a polynomial signature then the following defines the activation of $t : \mathbf{1} \to \mu(\Sigma \hat{+} F)^{||}$ for every Σ-algebra ϕ, by setting $G = \mathbf{I}$, initially.

Let $\Sigma, F, G : \mathcal{C} \to \mathcal{C}$ be polynomial functors, and $t : \mathbf{1} \to \mu(\Sigma \hat{+} F)^{||G}$. The arrow

$$\sqrt{t_\phi} : (\phi^{||} \leftarrow \phi^{||F}) \to \phi^{||G}$$

is defined, inductively on the structure of t, as follows:

- If the term consists of just one variable $x : \mathbf{1} \to X$, then $t = \mu(\Sigma \hat{+} F) \circ \iota_2 \circ x$, and

  $$\sqrt{t_\phi} = @ \circ (\mathbf{id} \vartriangle (x \circ !)).$$

- If $G = \mathbf{I}$ and t is not a variable, then, because $\mu(\Sigma \hat{+} F)$ is an isomorphism, $t = \mu(\Sigma \hat{+} F) \circ \iota_1 \circ t'$ for some $t' : \mathbf{1} \to \mu(\Sigma \hat{+} F)^{||\Sigma \hat{+} F}$. In this case,

  $$\sqrt{t_\phi} = \phi \circ \sqrt{t'_\phi}$$

  where the existence of $\sqrt{t'_\phi}$ is the induction hypothesis.

- If $G = \underline{A}$, and $t : \mathbf{1} \to A$,

  $$\sqrt{t_\phi} = t \circ !.$$

- If $G = G_1 \hat{\times} G_2$, $t = t_1 \vartriangle t_2$, and

$$\sqrt{t}_\phi = \sqrt{t_1}_\phi \vartriangle \sqrt{t_2}_\phi.$$

- If $G = G_1 \hat{+} G_2$ and $t = \iota_i \circ t'$,

$$\sqrt{t}_\phi = \iota_i \circ \sqrt{t}_\phi.$$

In certain base categories, like **CPO**, there is no coproduct, but only a weak coproduct, i.e. the arrow $f \triangledown g$ is not unique. That means that not all arrows $t : \mathbf{1} \to \mu(\Sigma \hat{+} F)^{\|G_1} + \mu(\Sigma \hat{+} F)^{\|G_2}$ can be written as $t = \iota_i \circ t'$. For all t where this is a problem, we must extend the above definition. For instance, in **CPO** we would define that for $t = \bot_{\mu(\Sigma \hat{+} F)^{\|G_1} + \mu(\Sigma \hat{+} F)^{\|G_2}}$,

$$\sqrt{t}_\phi = \bot_{(\phi^{\|G_1} + \phi^{\|G_2})} \circ !$$

□

The proof that this is inded an activation in the sense of definition 5 is a tedious exercise in writing out definitions.

4 Equivalences and congruences

In this section, we shall define the quotient of an algebra with respect to an equation. We shall make a slight generalization and use arrows like

$$p : \| \, ; R \dashrightarrow \|$$

instead of

$$\sqrt{t} : \| \hat{\leftarrow} (\| \, ; F) \dashrightarrow \|,$$

where $R = (\mathbf{I} \hat{\leftarrow} F)$.

Definition 5 (equivalence relation) An equivalence relation $p \simeq q$ is given by a pair

$$p \, , q : \| \, ; R \dashrightarrow \| \, , \quad \text{for some } R : \mathcal{C} \to \mathcal{C}.$$

The property that this equivalence relation holds in the Σ-algebra ϕ (ϕ satisfies $p \simeq q$) is defined as

$$\phi \models p \simeq q \;\Leftrightarrow\; p_\phi = q_\phi.$$

□

If we have an algebra ϕ which does not satisfy the equivalence $p \simeq q$, then we may form equivalence classes, and obtain a new algebra which does satisfy the equivalence. It is well known that in **Set** we can make equivalence classes of $\phi^{||}$ under $p \simeq q$ by taking the coequalizer of p and q. We generalize this to arbitrary base categories.

Definition 6 (equivalence quotient) The quotient for carrier objects in $\mathcal{C}$ w.r.t. the equivalence $p \simeq q$ is an endofunctor $/(p \simeq q)$ defined on the image of $||$ in $\mathcal{C}$, and a natural transformation

$$(p \simeq q) : || \dot{\rightarrow} || \,;\, /(p \simeq q).$$

The object-part of the functor $/(p \simeq q)$, and $(p \simeq q)$ are defined as the coequalizer of p and q. For a homomorphism $h : \phi \rightarrow \psi$, $h^{||/(p \simeq q)} : \phi^{||/(p\simeq q)} \rightarrow \psi^{||/(p\simeq q)}$ is the unique arrow that is given by the coequalizer for $(p \simeq q)_\psi \circ h^{||}$. This depends on the fact that $(p \simeq q)_\psi \circ h^{||} \circ p_\phi = (p \simeq q)_\psi \circ h^{||} \circ q_\phi$ which follows from the naturality of p and q.
□

This is illustrated with the following diagram:

$$\begin{array}{ccccc}
\phi^{||R} & \overset{p_\phi}{\underset{q_\phi}{\rightrightarrows}} & \phi^{||} & \xrightarrow{(p \simeq q)_\phi} & \phi^{||/(p\simeq q)} \\
\downarrow{\scriptstyle h^{||R}} & & \downarrow{\scriptstyle h^{||}} & & \vdots{\scriptstyle h^{||/(p\simeq q)}} \\
\psi^{||R} & \overset{p_\psi}{\underset{q_\psi}{\rightrightarrows}} & \psi^{||} & \xrightarrow[(p \simeq q)_\psi]{} & \psi^{||/(p\simeq q)}
\end{array}$$

The use of quotients of equivalence relations brings up some big problems. The above definition only makes quotients of carriers, not of complete algebras. Thus it does not give an algebra ϕ' such that $\phi'^{||} = \phi^{||/(p\simeq q)}$ and $\phi' \models p \simeq q$. Even if there is such an algebra, there is no guarantee that the arrow $(p \simeq q)_\phi : \phi^{||} \rightarrow \phi^{||/(p\simeq q)}$ can be extended into a homomorphism. This should come as no surprise for those who know about algebraic specifications. What we really need is a congruence relation instead of an equivalence relation.

In the standard theory of algebraic specifications, a congruence relation is an equivalence relation which is *compatible*. Compatibility means that the equivalence defined above is 'propagated within expressions', i.e. we do not only require

$$p_\phi = q_\phi$$

but also

$$\phi \circ {p_\phi}^{\Sigma} = \phi \circ {q_\phi}^{\Sigma} \,,$$
$$\phi \circ \phi^{\Sigma} \circ {p_\phi}^{\Sigma^2} = \phi \circ \phi^{\Sigma} \circ {q_\phi}^{\Sigma^2} ,$$

etcetera. Thus, we do not only quantify over all environments (in the case where an equation is given by a pair of terms with variables), but also over all *contexts* ϕ, $\phi \circ \phi^{\Sigma}$, $\phi \circ \phi^{\Sigma} \circ \phi^{\Sigma^2}$, etcetera. We should construct a kind of coequalizer which coequalizes all of these, i.e. all pairs of arrows

$$\phi \| R \Sigma^i \underset{q^{\Sigma^i}}{\overset{p^{\Sigma^i}}{\rightrightarrows}} \phi \| \Sigma^{i-1} \underset{\phi^{\Sigma^{i-1}}}{\overset{\phi^{\Sigma^{i-1}}}{\rightrightarrows}} \cdots \underset{\phi}{\overset{\phi}{\rightrightarrows}} \phi \|$$

This is done in the form of an ω-colimit, similar to the well known initial algebra construction. In order to make this work, we start with the coequalizer of

$$\mathbf{i}_{\phi\|}\,,\,\mathbf{i}_{\phi\|} : \mathbf{o} \to \phi\|$$

which is just the identity $\mathbf{id}_{\phi\|}$. Step by step, we add arrows, obtaining

$$\gamma_0 = \mathbf{i}_{\phi\|} \qquad \gamma_i = (\phi \circ {\gamma_{i-1}}^{\Sigma}) \triangledown p_{\phi}$$
$$\gamma'_0 = \mathbf{i}_{\phi\|} \qquad \gamma'_i = (\phi \circ {\gamma'_{i-1}}^{\Sigma}) \triangledown q_{\phi}$$

By the definition of the $(\Sigma \hat{+} \underline{\phi\|^R})$-algebras $\phi^{[p]}$ and $\phi^{[q]}$, and the fact that

$$(\phi \circ f^{\Sigma}) \triangledown \alpha_{\phi} = (\phi \triangledown \alpha_{\phi}) \circ (f^{\Sigma} + \mathbf{id}_{\phi\|R}) = \phi^{[\alpha]} \circ f^{(\Sigma \hat{+} \underline{\phi\|^R})}$$

this may also be written as

$$\gamma_0 = \mathbf{i}_{\phi\|} \qquad \gamma_i = \phi^{[p]} \circ {\gamma_{i-1}}^{\Sigma \hat{+} \underline{\phi\|^R}} : \mathbf{o}^{(\Sigma \hat{+} \underline{\phi\|^R})^i} \to \phi^{[p]}\|\|$$
$$\gamma'_0 = \mathbf{i}_{\phi\|} \qquad \gamma'_i = \phi^{[q]} \circ {\gamma'_{i-1}}^{\Sigma \hat{+} \underline{\phi\|^R}} : \mathbf{o}^{(\Sigma \hat{+} \underline{\phi\|^R})^i} \to \phi^{[q]}\|\|.$$

This gives two cocones γ and γ' on the ω-chain $\left(\mathbf{o}^{\Sigma \hat{+} \underline{\phi\|^R}}\right)_{i \geq 0}$. The ω-colimit of this chain is $\mu(\Sigma \hat{+} \underline{\phi\|^R})$, and the colimits of the cones γ and γ' are $(\!|\phi^{[p]}|\!)$ and $(\!|\phi^{[q]}|\!)$.

The result of this somewhat complex argument is summarized in the following definition.

Definition 7 (congruence relation) A congruence relation $p \cong q$ is given by a pair

$$p\,,q : \| \,;\, R \dashrightarrow \|\,, \quad \text{for some } R : \mathcal{C} \to \mathcal{C}.$$

The fact that this congruence relation holds in the Σ-algebra ϕ (ϕ satisfies $p \cong q$) is defined as

$$\phi \models p \cong q \;\Leftrightarrow\; \mathbf{i}_{\phi^{[p]}} = \mathbf{i}_{\phi^{[q]}}.$$

This is equivalent to saying that

$$(\!|\phi^{[p]}|\!)\,,\,(\!|\phi^{[q]}|\!) : (\phi\|^R)^{\mathcal{F}_{\Sigma}\|} \to \phi\|$$

are equal.
□

We can define quotients w.r.t. congruence relations similarly to definition 6.

Definition 8 (congruence quotient) The quotient in $\mathcal{C}|\Sigma$ w.r.t. the congruence $p \cong q$, where $p, q : \| ; R \dashrightarrow \|$, is an endofunctor $/(p \cong q) : \mathcal{C}|\Sigma \longrightarrow \mathcal{C}|\Sigma$ and a natural transformation

$$(p \cong q) : \mathbf{I} \dashrightarrow /(p \cong q).$$

The object-part of the functor $/(p \cong q)$, and $(p \cong q)$ are defined as the coequalizer of the parallel pair

$$\phi^{\|R\mathcal{F}_\Sigma} \underset{(\mathsf{i}_{(\phi[q])})^{\backslash(\underline{\phi\|R})}}{\overset{(\mathsf{i}_{(\phi[p])})^{\backslash(\underline{\phi\|R})}}{\rightrightarrows}} \phi$$

in the category $\mathcal{C}|\Sigma$. For a homomorphism $h : \phi \longrightarrow \psi$,

$$h^{\|/(p\cong q)} : \phi^{\|/(p\cong q)} \longrightarrow \psi^{\|/(p\cong q)}$$

is the unique arrow that is given by the coequalizer for $(p \cong q)_\psi \circ h$.
□

Note that the quotient is only defined if the above coequalizer exists in $\mathcal{C}|\Sigma$.
It is easy to see that definition 8 does what we want.

Proposition 6

$$\phi^{/(p\cong q)} \models p \cong q$$

□

Proof

$$\begin{array}{ll} & \phi^{/(p\cong q)} \models p \cong q \\ \Leftrightarrow & \text{(definition 7)} \\ & \mathsf{i}_{\phi/(p\cong q)[p]} = \mathsf{i}_{\phi/(p\cong q)[q]} \\ \Leftrightarrow & \text{(initiality)} \\ & (p \cong q)_\phi \circ \mathsf{i}_{\phi[p]} = (p \cong q)_\phi \circ \mathsf{i}_{\phi[q]} \\ \Leftarrow & \text{(coequalizer)} \end{array}$$

□

The following proposition says that for algebras, satisfaction of a congruence and satisfaction of an equivalence are the same.

Proposition 7

$$\phi \models p \cong q \;\Leftrightarrow\; \phi \models p \simeq q$$

□

Proof

$$
\begin{aligned}
& \mathbf{i}_{\phi[p]} = \mathbf{i}_{\phi[q]} \\
\Leftrightarrow\; & \phi^{[p]} = \phi^{[q]} \\
\Leftrightarrow\; & \phi \triangledown p_\phi = \phi \triangledown q_\phi \\
\Leftrightarrow\; & p_\phi = q_\phi
\end{aligned}
$$

□

By taking the quotient of an algebra, we have obtained the intended meaning of equations $t = t'$.

Proposition 8

$$\phi \models \surd t \cong \surd t' \;\Leftrightarrow\; \forall \alpha : \underline{X} \dashrightarrow \| : \phi^{[\alpha]} \models t \sim t'$$

□

Proof

$$
\begin{aligned}
& \mathbf{i}_{\phi[\surd t]} = \mathbf{i}_{\phi[\surd t']} \\
\Leftrightarrow\; & \text{(proposition 7)} \\
& \surd t_\phi = \surd t'_\phi \\
\Leftrightarrow\; & \text{(extensionality)} \\
& \forall \alpha : \underline{X} \dashrightarrow \| : \surd t_\phi \circ \text{`}\alpha_\phi\text{'} = \surd t'_\phi \circ \text{`}\alpha_\phi\text{'} \\
\Leftrightarrow\; & \text{(definition of } \surd\text{)} \\
& \forall \alpha : \underline{X} \dashrightarrow \| : (\!|\phi^{[\alpha]}|\!) \circ t = (\!|\phi^{[\alpha]}|\!) \circ t'
\end{aligned}
$$

□

The formalism developed here can also be used to handle systems of multiple equations. The meaning of such a system is defined by

$$\phi \models (p_1 \cong q_1 \wedge \cdots \wedge p_n \cong q_n) \;\Leftrightarrow\; \phi \models p_1 \cong q_1 \wedge \cdots \wedge \phi \models p_n \cong q_n$$

where p_i , q_i : $\|$; $R_i \dashrightarrow \|$. By definition 7, this is equivalent to

$$\mathbf{i}_{\phi[p_1]} = \mathbf{i}_{\phi[q_1]} \wedge \cdots \wedge \mathbf{i}_{\phi[p_n]} = \mathbf{i}_{\phi[q_n]}.$$

Definition 2 says that $\phi^{[p_i]} = \phi \triangledown p_{i_\phi}$. These algebras have ϕ in common, so we can combine them into a bigger algebra

$$\phi^{[p_1]\cdots[p_n]} = \phi \triangledown p_{1_\phi} \triangledown \cdots \triangledown p_{n_\phi} = \phi^{[p_1 \triangledown \cdots \triangledown p_n]} : \phi^{\|\Sigma} + \phi^{\|R_1} + \cdots + \phi^{\|R_n} \to \phi^{\|}.$$

This leads to the following proposition.

Proposition 9

$$\phi \models (p_1 \cong q_1 \wedge \cdots \wedge p_n \cong q_n) \;\Leftrightarrow\; \phi \models (p_1 \triangledown \cdots \triangledown p_n) \cong (q_1 \triangledown \cdots \triangledown q_n)$$

□

Proof

$$\begin{aligned} & \mathfrak{i}_{\phi[p_1]} = \mathfrak{i}_{\phi[q_1]} \wedge \cdots \wedge \mathfrak{i}_{\phi[p_n]} = \mathfrak{i}_{\phi[q_n]} \\ \Leftrightarrow\; & \text{(apply the proof of proposition 7)} \\ & p_{1\phi} = q_{1\phi} \wedge \cdots \wedge p_{n\phi} = q_{n\phi} \\ \Leftrightarrow\; & \mathfrak{i}_{\phi[p_1 \triangledown \cdots \triangledown p_n]} = \mathfrak{i}_{\phi[q_1 \triangledown \cdots \triangledown q_n]}. \end{aligned}$$

□

This proposition means that we may transform a system of multiple equations into one equation. Therefore such systems can be handled without difficulty.

5 Constructing quotients

As we saw in the previous section, taking the quotient algebra of a Σ-algebra under some congruence relation involves a coequalizer in the category $\mathcal{C}|\Sigma$. It is clear that this coequalizer need not exist for all possible congruence relations. Diaconescu gives a general construction in [2], but he *assumes* the existence of coequalizers (and pullbacks) in $\mathcal{C}|\Sigma$. Here, we would rather use properties of $\mathcal{C}$, which is the category in which things are actually 'computed'. Using these properties, we might *construct* quotient algebras for certain congruence relations.

The standard set-theoretic construction of congruences uses an axiom of choice, which says that we may pick representatives from congruence classes. The category theoretical version of this is formulated as follows (Diaconescu [2] has another variant).

Definition 9 An epimorphism $e : A \to B$ *splits* if there exists an arrow $r : B \to A$, such that $e \circ r = \mathbf{id}_B$.
□

Definition 10 (compatible equivalence) The equivalence quotient $(p \simeq q)$ (see definition 6) is *compatible* w.r.t. the algebra ϕ if for all objects A, and all arrows $x\,, y : A \to \phi^{||\Sigma}$,

$$(p \simeq q)^{\Sigma} \circ x = (p \simeq q)^{\Sigma} \circ y \;\Rightarrow\; (p \simeq q) \circ \phi \circ x = (p \simeq q) \circ \phi \circ y.$$

□

Proposition 10 The congruence quotient $(p \cong q)$ viewed as an equivalence quotient $(p \cong q)^{||}$ is compatible for every algebra.
□

Proof

$$
\begin{array}{ll}
 & (p \cong q)^{||\Sigma} \circ x = (p \cong q)^{||\Sigma} \circ y \\
\Rightarrow & \\
 & \phi^{/(p\cong q)} \circ (p \cong q)^{||\Sigma} \circ x = \phi^{/(p\cong q)} \circ (p \cong q)^{||\Sigma} \circ y \\
\Leftrightarrow & ((p \cong q) \text{ is a homomorphism}) \\
 & (p \cong q)^{||} \circ \phi \circ x = (p \cong q)^{||} \circ \phi \circ y
\end{array}
$$

□

Proposition 11 An equivalence quotient which splits and is compatible can be extended into a congruence quotient.
□

Proof Suppose that $(p \cong q)$ splits (it is epi because it is the arrow of a coequalizer) as $(p \cong q) \circ r = \mathbf{id}$. We define the congruence quotient as

$$
\begin{array}{l}
(p \cong q) = (p \simeq q) \\
\phi^{/(p\cong q)} = (p \simeq q)_\phi \circ \phi \circ {r_\phi}^\Sigma
\end{array}
$$

All we must do now is prove that $(p \cong q)$ is a homomorphism:

$$
\begin{array}{ll}
 & \phi^{/(p\cong q)} \circ (p \cong q)^\Sigma = (p \cong q) \circ \phi \\
\Leftrightarrow & (\text{definition of } \phi^{/(p\cong q)}, (p \cong q)) \\
 & (p \simeq q) \circ \phi \circ {r_\phi}^\Sigma \circ (p \simeq q)^\Sigma = (p \simeq q) \circ \phi \\
\Leftarrow & (\text{compatibility}) \\
 & (p \simeq q)^\Sigma \circ {r_\phi}^\Sigma \circ (p \simeq q)^\Sigma = (p \simeq q)^\Sigma \\
\Leftarrow & ((p \simeq q) \text{ splits}) \\
 & (p \simeq q)^\Sigma = (p \simeq q)^\Sigma
\end{array}
$$

□

In a practical implementation, the congruence classes are computed by a rewriting system, which is what the arrow $(p \cong q)$ models. The more powerful the rewriting system, the more coequalizers we can compute. The simplest rewriting system is one for simple function definitions of the form $f(x) = \mathit{expr}$. This is implemented in every functional programming language, and we should be able to do at least this in the categorical framework.

The left hand side of the equation $f(x) = expr$ is the application of one of the operators from the signature, f, to some variables, x. We assume that $\Sigma = \Sigma_1 \hat{+} \Sigma_2$, where Σ_2 is that part of the signature that corresponds to f. The left hand side is now a constant

$$\ell : \mathbf{1} \to \mu(\Sigma \hat{+} \underline{X})^{||}$$
$$\ell = \mathbf{1} \xrightarrow{x} X^{\Sigma_2} \xrightarrow{(\mu(\Sigma \hat{+} \underline{X}) \circ \iota_2)^{\Sigma_2}} \mu(\Sigma \hat{+} \underline{X})^{||\Sigma_2} \xrightarrow{\mu(\Sigma \hat{+} \underline{X}) \circ \iota_1 \circ \iota_2} \mu(\Sigma \hat{+} \underline{X})^{||}$$

We must impose the condition that all variables in X occur in the left hand side, i.e. there are no variables in *expr* that do not occur in x. The variables are given by X, and $\phi^{||} \leftarrow X$ are all possible ϕ-assignments to these variables. The inputs for f in ϕ are given by ϕ^{Σ_2}, so Σ_2 corresponds to the variables that occur in the left hand side. This condition then becomes

$$(\leftarrow X) = \Sigma_2.$$

When this condition is satisfied,

$$\sqrt{\ell}_\phi = \phi \circ \iota_2 : \phi^{||\Sigma_2} \to \phi^{||}.$$

For the right hand side, we require that f does not occur in it. This means that we do not allow recursive definitions; we only allow expressions from which recursive calls have been removed, or 'canned' recursion, in the form of catamorphisms. The right hand side now becomes

$$r : \mathbf{1} \to \mu(\Sigma_1 \hat{+} \underline{X})^{||}$$

and $\sqrt{r}$ is defined for Σ_1-algebras (including $\phi^{\backslash \Sigma_2}$ for any Σ-algebra ϕ).

Proposition 12 The quotient of $\mu(\Sigma)$ w.r.t. the congruence relation $\sqrt{\ell} \cong \sqrt{r}$ is

$$\mu(\Sigma)^{/(\sqrt{\ell} \cong \sqrt{r})} = \mu(\Sigma_1)^{[\sqrt{r}]}$$
$$(\sqrt{\ell} \cong \sqrt{r})_{\mu(\Sigma)} = \mathbf{i}_{(\mu(\Sigma_1)^{[\sqrt{r}]})} : \mu(\Sigma) \to \mu(\Sigma_1)^{[\sqrt{r}]}$$

□

The homomorphism $(\sqrt{\ell} \cong \sqrt{r})_{\mu(\Sigma)}$ replaces occurrences of $f(x)$ by *expr* throughout the term. This corresponds to the semantics of functional languages. The carrier of the algebra $\mu(\Sigma_1)$ contains all expressions that are in normal form.

Proof We must show that

$$\mu(\Sigma)^{||\Sigma_2 \mathcal{F}_\Sigma} \underset{(\mathbf{i}_{\mu(\Sigma)[\sqrt{r}]})^{\backslash(\underline{\mu(\Sigma)^{||\Sigma_2}})}}{\overset{(\mathbf{i}_{\mu(\Sigma)[\sqrt{\ell}]})^{\backslash(\underline{\mu(\Sigma)^{||\Sigma_2}})}}{\rightrightarrows}} \mu(\Sigma) \xrightarrow{\mathbf{i}_{(\mu(\Sigma_1)^{[\sqrt{r}]})}} \mu(\Sigma_1)^{[\sqrt{r}]}$$

is a coequalizer, as required by definition 8. First, we show that the two arrows are equal:

$$
\begin{array}{cl}
 & \mathfrak{i}_{(\mu(\Sigma_1)^{[\sqrt{r}]})} \circ \left(\mathfrak{i}_{\mu(\Sigma)^{[\sqrt{\ell}]}}\right)^{\backslash(\underline{\mu(\Sigma)^{\|\Sigma_2}})} \\
= & \text{(proposition 2)} \\
 & \left(\mathfrak{i}_{\mu(\Sigma_1)^{[\sqrt{r}][\sqrt{\ell}]}}\right)^{\backslash(\underline{\mu(\Sigma)^{\|\Sigma_2}})} \\
= & \text{(from the definition of } \sqrt{\ell}, \\
 & \qquad \sqrt{\ell}_{(\mu(\Sigma_1)^{[\sqrt{r}]})} = (\mu(\Sigma_1) \triangledown \sqrt{r}_{(\mu(\Sigma_1)^{[\sqrt{r}]})}) \circ \iota_2 = \sqrt{r}_{(\mu(\Sigma_1)^{[\sqrt{r}]})} \\
 &) \\
 & \left(\mathfrak{i}_{\mu(\Sigma_1)^{[\sqrt{r}][\sqrt{r}]}}\right)^{\backslash(\underline{\mu(\Sigma)^{\|\Sigma_2}})} \\
= & \text{(proposition 2)} \\
 & \mathfrak{i}_{(\mu(\Sigma_1)^{[\sqrt{r}]})} \circ \left(\mathfrak{i}_{\mu(\Sigma)^{[\sqrt{r}]}}\right)^{\backslash(\underline{\mu(\Sigma)^{\|\Sigma_2}})}
\end{array}
$$

Next we prove that there is a unique homomorphism to any other algebra ϕ, for which

$$
\begin{array}{cl}
 & \mathfrak{i}_\phi \circ \left(\mathfrak{i}_{\mu(\Sigma)^{[\sqrt{\ell}]}}\right)^{\backslash(\underline{\mu(\Sigma)^{\|\Sigma_2}})} = \mathfrak{i}_\phi \circ \left(\mathfrak{i}_{\mu(\Sigma)^{[\sqrt{r}]}}\right)^{\backslash(\underline{\mu(\Sigma)^{\|\Sigma_2}})} \\
\Leftrightarrow & \text{(proposition 2)} \\
 & \left(\mathfrak{i}_{\phi[\sqrt{\ell}]}\right)^{\backslash(\underline{\mu(\Sigma)^{\|\Sigma_2}})} = \left(\mathfrak{i}_{\phi[\sqrt{r}]}\right)^{\backslash(\underline{\mu(\Sigma)^{\|\Sigma_2}})} \\
\Leftrightarrow & \text{(homomorphism determined by carrier)} \\
 & (\!|\phi^{[\sqrt{\ell}]}|\!) = (\!|\phi^{[\sqrt{r}]}|\!) \\
\Leftrightarrow & \text{(catamorphism determined by algebra)} \\
 & \sqrt{\ell}_\phi = \sqrt{r}_\phi .
\end{array}
$$

By definition of $\sqrt{\ell}$, proposition 1, and the above line,

$$\phi = \phi^{\backslash\Sigma_2[\sqrt{\ell}]} = \phi^{\backslash\Sigma_2[\sqrt{r}]}.$$

By initiality of $\mathfrak{i}_\phi$,

$$\mathfrak{i}_\phi = (\mathfrak{i}_{\phi\backslash\Sigma_2})^{[\sqrt{r}]} \circ \mathfrak{i}_{(\mu(\Sigma_1)^{[\sqrt{r}]})}$$

and $(\mathfrak{i}_{\phi\backslash\Sigma_2})^{[\sqrt{r}]}$ is the unique arrow such that this holds.
□

Proposition 12 only defines the quotient of the initial algebra. Since it makes critical use of catamorphisms, it is not easily extendable to general algebras. In many cases, this is no problem. For instance, the meaning of a functional program given by a set of function definitions is given by the quotient of the initial algebra w.r.t. the system of multiple equations arising from these function definitions. In proposition 9 we saw how to handle such a system. In this case, it means choosing Σ_2 such that it incorporates all the defined function symbols.

6 Discussion

Although we did not mention them, many-sorted algebras and algebras with parameters fit in easily with our description. Not all aspects of quotient algebras have been investigated in this paper. We have only looked at systems of multiple equations, but not at more complex ways of combining equations. Ideally, all boolean connectives should be allowed in equations, but the theory presented here is not capable to describe this.

It is rather disturbing that initial algebras play such an important role in contexts (see page 11). After having considered several other ways of specifying contexts, the way we handled them here still appeared to be the best alternative.

Probably the most important result is that we have a simple characterization of congruences and quotients (c.f. definitions 7 and 8), which is applicable in practical situations (c.f. propositions 5 and 12).

References

[1] Michael Barr, Charles Wells, *Category Theory for Computing Science*, Prentice Hall 1990.

[2] Răzvan Diaconescu, *The Formal Completeness of Equational Logics*, Programming Research Group, Oxford University, August 1991.

[3] H. Ehrig, B. Mahr, *Fundamentals of Algebraic Specification 1, Equations and Initial Semantics*, Springer-Verlag 1985.

[4] Maarten M. Fokkinga, *Law and Order in Algorithmics*, thesis.

[5] Benjamin C. Pierce, *A Taste of Category Theory for Computer Scientists*, tech. rep. CMU-CS-90-113, Carnegie Mellon University, 1990.

[6] Kristoffer H. Rose, *Typesetting Diagrams with XY-pic: User's Manual*, DIKU, 1992. (This manual and the TeX macro package it describes are available via FTP from `diku.dk` and `archive.cs.ruu.nl`.)

[7] D.E. Rydeheard, R.M. Burstall, *Computational Category Theory*, Prentice Hall 1988.

[8] Martin Wirsing, *Algebraic Specification*, in Jan van Leeuwen, ed., *Handbook of Theoretical Computer Science, volume B, Formal Models and Semantics*, Elsevier/MIT press 1990.

This is illustrated in the following diagram.

$$\begin{array}{ccccc}
\mu(\Sigma \hat{+} \underline{A})^{\|\Sigma} + A & & \xrightarrow{\mu(\Sigma \hat{+} \underline{A})} & & \mu(\Sigma \hat{+} \underline{A})^{\|} \\
\Big\downarrow {\scriptstyle (\!|(\mu(\Sigma \hat{+} \underline{B}) \circ (\mathbf{id}+f))|\!)^{\Sigma} + \mathbf{id}_A} & & & & \Big\downarrow {\scriptstyle (\!|(\mu(\Sigma \hat{+} \underline{B}) \circ (\mathbf{id}+f))|\!)} \\
\mu(\Sigma \hat{+} \underline{B})^{\|\Sigma} + A & \xrightarrow[\mathbf{id}+f]{} & \mu(\Sigma \hat{+} \underline{B})^{\|\Sigma} + B & \xrightarrow[\mu(\Sigma \hat{+} \underline{B})]{} & \mu(\Sigma \hat{+} \underline{B})^{\|}
\end{array}$$

The algebra $A^{\mathcal{F}_\Sigma}$ is called the free Σ-algebra over A.

Proof It is easy to check that this indeed defines a functor from $\mathcal{C}$ to $\mathcal{C}|\Sigma$. Since

$$A^{\mathcal{F}_\Sigma \|} = \mu(\Sigma \hat{+} \underline{A})^{\backslash \underline{A} \|} = \mu(\Sigma \hat{+} \underline{A})^{\|}$$

we may define for every $A \in \mathcal{C}_o$ an arrow

$$\eta_A = \mu(\Sigma \hat{+} \underline{A}) \circ \iota_2 : A \to \mu(\Sigma \hat{+} \underline{A})^{\|}$$

Now for every $f : A \to \phi^{\|}$ there is a unique arrow

$$\mathbf{i}_{\phi \triangledown f_\phi} : \mu(\Sigma \hat{+} \underline{A}) \to \phi \triangledown f_\phi$$

such that

$$\mathbf{i}_{\phi \triangledown f_\phi}{}^{\|} \circ \eta_A = f.$$

□

The intended meaning of an equation $t = t'$ where variables from X occur in both sides, is that for all assignments to X the interpretations of these terms are equal. That is, we require

$$\forall \alpha : \underline{X} \dashrightarrow \| : \phi^{[\alpha]} \models t \sim t'.$$

The problem with this is that we have a different algebra $\phi^{[\alpha]}$ for every assignment α, so we cannot say what it means that the algebra ϕ itself satisfies $t = t'$. In order to do this, we must bring the quantification over α 'inside'. This is done by parameterizing the terms in the equation with α, thus making the quantification implicit.

Definition 4 (activation of terms) An *activation* in a Σ-algebra ϕ of a term

$$t : \mathbf{1} \to \mu(\Sigma \hat{+} F)^{\|}$$

is an arrow

$$\sqrt{t}_\phi : (\phi^{\|} \leftarrow \phi^{\|F}) \to \phi^{\|}$$

such that for all $\alpha : \| \,; F \dashrightarrow \|$

$$\sqrt{t}_\phi \circ `\alpha_\phi' = (\!|\phi^{[\alpha]}|\!) \circ t.$$

□