Congruences on Initial Algebras

Roland C. Backhouse,
Department of Mathematics and Computing Science,
Eindhoven University of Technology,
P.O. Box 513,
5600 MB Eindhoven,
The Netherlands.

Grant Malcolm,
Department of Computing Science,
Rijksuniversiteit Groningen,
P.O. Box 800,
9700 AV Groningen,
The Netherlands.

Jaap van der Woude,
Department of Mathematics and Computing Science,
Eindhoven University of Technology,
P.O. Box 513,
5600 MB Eindhoven,
The Netherlands.

September 4, 1990

Abstract

The notion of relational catamorphism is used to give a necessary and sufficient condition for a relation to be a congruence relation on an abstract data type. The condition is shown to be a generalisation of the homomorphism theorem of universal algebra.

Initial algebra semantics provides an attractive formalism for defining abstract data types (see [5, 11, 10]). An advantage of the formalism is that initiality gives a mechanism for constructing and proving equality of recursive functions on the defined type: the algebraic properties of these functions can be exploited in developing a calculus of program transformation (see [8, 9]). Abstract data types so defined have a particularly elegant semantics in the category of sets with total functions as arrows; in a lecture given by the first-named author [1], it was shown that, by restricting the notion of functor, data types which are initial algebras on such a category are also initial on an underlying category of sets with relations as arrows. In this setting, initiality provides a mechanism for constructing recursive relations, thereby generalising the construction of recursive functions and allowing non-determinism. The algebraic properties of these relations lead to a calculus of program refinement, a topic to be set out in detail in a forthcoming paper [2].

Beside the desire to allow non-determinism, a further motivation to study abstract data types in a relational setting is to provide a homogeneous treatment of types with laws. Essentially, one replaces the equality relation on the underlying type with a congruence relation: for example, the type of binary trees, modulo associativity of the node operator, gives the type of semigroups.

This paper contains a preliminary investigation of congruence relations on initial algebra data structures. The main result is a necessary and sufficient condition for a relation to be a congruence relation. A related result, given in the final section, is a proof of a theorem cited by Ehrig & Mahr [4]: the theorem states that a relation constructed from a function is a congruence relation iff the function is a homomorphism. We show that the theorem is a corollary to our characterisation of congruence relations.

Notation is introduced and concepts defined in section 1 and the first half of section 2. These should not be considered mere preparation, for we are just as interested in *how we prove* as in *what we prove*. Our broader interest lies in developing a type-oriented calculus for the formal manipulation of relations (and, therefore, of programs). This is why we prefer the initial algebra characterisation of data types to the inductive characterisation (as in, for example, Martin-Löf's type theory).

The exposition of relations below is couched in set-theoretic terms; the paper referred to above ([2]) presents an axiomatic theory which has the set-theoretic relations as a model.

1 Relations

That R is a relation between objects of type α and objects of type β is denoted by $R \in \alpha \sim \beta$. Given this notation, there is only one sensible notation for composition of relations: the composition of $R \in \alpha \sim \beta$ and $S \in \beta \sim \gamma$ is written

$$R \circ S \in \alpha \sim \gamma,$$

and has the standard meaning that $a \in \alpha$ is related by the composition to $c \in \gamma$ iff there is a $b \in \beta$ such that R relates a to b and S relates b to c.

The equality relation for a type α is denoted by α itself, so that

$$R \in \alpha \sim \beta \quad \equiv \quad \alpha \circ R = R = R \circ \beta.$$

Composition is associative, and each type has an equality relation which is a neutral element of composition, so we have just described a category, whose objects are types and whose arrows are relations. In particular, the identity arrow for a given type is that type's equality relation. This simple structure is further enriched by the standard operations on set-theoretic relations: intersection ($\cap$), union ($\cup$), containment ($\supseteq$), and converse ($^{\cup}$). We write converse as a postfix operator, thus the converse of $R \in \alpha \sim \beta$ is $R^{\cup} \in \beta \sim \alpha$. A particularly useful property of converse is its distribution through composition:

$$(R \circ S)^{\cup} \quad = \quad S^{\cup} \circ R^{\cup}.$$

Note the reversal of R and S.

Functions are considered to be special cases of relations: that is, function $f \in \alpha \leftarrow \beta$ is also a relation of type $\alpha \sim \beta$ such that $a \in \alpha$ stands in relation f to $b \in \beta$ iff $a = f.b$. For example, the identity function on a type is just the equality relation on that type. We use the notation $\alpha \leftarrow \beta$ for the type of total functions with range α and domain β. We also use lowercase letters $f, g, h, \ldots$ to denote (total) functions, and uppercase letters $R, S, T, \ldots$ to denote arbitrary relations. The defining characteristics of total functions can be expressed very neatly:

Definition 1 (total functions) That $R \in \alpha \sim \beta$ is functional is expressed by:

(function) $\quad\alpha \supseteq R \circ R^{\cup}$

and that it is total (defined everywhere on β) by:

(totality) $\quad R^{\cup} \circ R \supseteq \beta.$

Moreover, R is injective iff $R^{\cup}$ is functional, and R is surjective iff $R^{\cup}$ is total.
□

There are many properties specific to total functions, for example, that composition with functions distributes backwards through intersection. The following Galois correspondences, which the reader can easily prove for himself, are particularly useful.

Property 2 Suppose $R \in \alpha \sim \beta$, $S \in \alpha \sim \gamma$, and f is a total function of type $\beta \leftarrow \gamma$. Then

$$R \circ f \supseteq S \quad\equiv\quad R \supseteq S \circ f^{\cup}$$

□

Property 3 Suppose $R \in \alpha \sim \beta$, $S \in \gamma \sim \beta$, and f is a total function of type $\alpha \leftarrow \gamma$. Then

$$R \supseteq f \circ S \quad\equiv\quad f^{\cup} \circ R \supseteq S$$

□

2 Congruence Relations

We are interested in examining the properties of congruence relations on recursively defined types. The paradigm of type definition that we consider is based on that of Hagino types [6] and is closely related to initial algebra semantics of ADT's (see Manes and Arbib [10]); the paradigm extends both of these approaches to a relational setting. A data type is viewed as initial in a category of algebras whose structure is determined by a relator:

Definition 4 (relator) A *relator* maps types to types and relations to relations. Specifically, Φ is a relator iff:

(a) if α is a type, then $\Phi.\alpha$ is a type;

(b) if $R \in \alpha \sim \beta$ is a relation, then there is a relation $\Phi.R \in \Phi.\alpha \sim \Phi.\beta$;

(c) Φ preserves composition: $\Phi.(R \circ S) = \Phi.R \circ \Phi.S$;

(d) Φ preserves identity: that is, Φ maps the equality relation of type α to the equality relation of type $\Phi.\alpha$ — in our notation this takes the form of the truism $\Phi.\alpha = \Phi.\alpha$.

So far, these requirements state only that a relator is an endofunctor on the category of relations. A relator must also respect the extra structure of relations:

(e) Φ is monotonic: $R \supseteq S \Rightarrow \Phi.R \supseteq \Phi.S$; and

(f) Φ preserves converse: $\Phi.(R^{\cup}) = (\Phi.R)^{\cup}$ — in view of this equality, we shall omit the parentheses, writing simply $\Phi.R^{\cup}$.

□

We remark that these properties also guarantee that Φ preserves functions. Specifically, if $f \in \alpha \leftarrow \beta$ is a total function then $\Phi.f \in \Phi.\alpha \leftarrow \Phi.\beta$. Thus a relator is also a functor of the category **SET** with sets as objects and functions as arrows.

Definition 5 (initial types) For a relator Φ, the type $\mu\Phi$ is characterised (up to isomorphism) by:

(a) a bijection $\tau \in \mu\Phi{\leftarrow}\Phi.\mu\Phi$; and

(b) the "unique extension property" that for all $S \in \alpha \sim \Phi.\alpha$ there is a relation $(\!|S|\!) \in \alpha \sim \mu\Phi$ such that for all $R \in \alpha \sim \mu\Phi$,

$$\begin{array}{rcl} R = (\!|S|\!) & \equiv & R \circ \tau \;=\; S \circ \Phi.R \\ R \supseteq (\!|S|\!) & \Leftarrow & R \circ \tau \;\supseteq\; S \circ \Phi.R \\ R \subseteq (\!|S|\!) & \Leftarrow & R \circ \tau \;\subseteq\; S \circ \Phi.R \end{array}$$

□

The first of the three properties in (b) is the statement of initiality: for every Φ-algebra $(\alpha,\ S \in \alpha \sim \Phi.\alpha)$ there is precisely one homomorphism $(\!|S|\!) \in \alpha \sim \mu\Phi$. (A homomorphism from a Φ-algebra $(\beta,\ T \in \beta \sim \Phi.\beta)$ to another Φ-algebra $(\alpha,\ S \in \alpha \sim \Phi.\alpha)$ is a relation $H \in \alpha \sim \beta$ such that $H \circ T = S \circ \Phi.H$.) We shall refer to homomorphisms of the form $(\!|S|\!)$ as "catamorphisms". The other two properties in (b) arise from the fact that $(\!|S|\!)$ is the unique solution to

$$X \quad :: \quad X \;=\; S \circ \Phi.X \circ \tau\cup .$$

Thus, by Knaster-Tarski, it is also the least solution to the containment and the greatest solution to the inclusion. Proofs of these claims are given in [2].

We turn now to congruence relations on initial types.

Definition 6 (congruence relation) Relation $R \in \mu\Phi \sim \mu\Phi$ is a congruence relation iff it is an equivalence relation (i.e., it enjoys the familiar properties of reflexivity, transitivity and symmetry) and it is Φ-*substitutive*:

(refl.) $\quad R \supseteq \mu\Phi$

(trans.) $\quad R \supseteq R \circ R$

(symm.) $\quad R = R\cup$

(subst.) $\quad R \circ \tau \supseteq \tau \circ \Phi.R.$

We shall say that relation R is a *half*-congruence iff it is reflexive, transitive and substitutive (but not necessarily symmetric).

□

The remainder of this section contains a proof that a relation is a congruence relation iff it can be expressed in a certain form. We need one more definition:

Definition 7 For a relation $R \in \alpha \sim \beta$, the relation $R\dagger \in \beta \sim \beta$ is defined by the following property: for all $S \in \beta \sim \beta$,

$$R\dagger \supseteq S \quad \equiv \quad R \supseteq R \circ S.$$

□

The relation $R\dagger$ is a typed variant of the "weakest prespecification" R/R of Hoare and He (see [7]); its equational presentation lends itself well to the sort of calculational style of proof in which we are interested. (To be precise $R\dagger \;=\; \beta \circ R/R \circ \beta$. Strictly, therefore, the type β should appear as an argument to $\dagger$. We only use the definition here in the case that $\beta = \mu\Phi$.)

For the moment, we shall ignore symmetry and concentrate on the properties of half-congruences. Consider relation $R \in \mu\Phi \sim \mu\Phi$.

Property 8 $R\dagger$ is reflexive.
Proof: $R \supseteq R \circ \mu\Phi$, hence by definition 7, $R\dagger \supseteq \mu\Phi$.
□

Property 9 (cancellation) $R \supseteq R \circ R\dagger$.
Proof: $R\dagger \supseteq R\dagger$, hence by definition 7, $R \supseteq R \circ R\dagger$.
□

Property 10 $R\dagger$ is transitive.
Proof:

$$\begin{array}{ll}
 & R\dagger \supseteq R\dagger \circ R\dagger \\
\equiv & \quad \{ \text{definition 7} \} \\
 & R \supseteq R \circ R\dagger \circ R\dagger \\
\Leftarrow & \quad \{ \text{property 9, twice} \} \\
 & R \supseteq R \\
\equiv & \\
 & \mathsf{true}
\end{array}$$

□

Property 11 If R is a catamorphism, then $R\dagger$ is Φ-substitutive.
Proof: Assume that R is a catamorphism; in particular, let $R = (\!|S|\!)$, and therefore, by the unique extension property 5(b), $R \circ \tau = S \circ \Phi.R$. Substitutivity is proven as follows.

$$\begin{array}{ll}
 & R\dagger \circ \tau \;\supseteq\; \tau \circ \Phi.(R\dagger) \\
\equiv & \quad \{ \text{ property 2 } \} \\
 & R\dagger \;\supseteq\; \tau \circ \Phi.(R\dagger) \circ \tau^{\cup} \\
\equiv & \quad \{ \text{ definition 7 } \} \\
 & R \;\supseteq\; R \circ \tau \circ \Phi.(R\dagger) \circ \tau^{\cup} \\
\equiv & \quad \{ \text{ uep 5(b) } \} \\
 & R \;\supseteq\; S \circ \Phi.R \circ \Phi.(R\dagger) \circ \tau^{\cup} \\
\equiv & \quad \{ \text{ definition of relator, in particular 4(c) } \} \\
 & R \;\supseteq\; S \circ \Phi.(R \circ R\dagger) \circ \tau^{\cup} \\
\Leftarrow & \quad \{ \text{ property 9; } \Phi \text{ is monotonic } \} \\
 & R \;\supseteq\; S \circ \Phi.R \circ \tau^{\cup} \\
\equiv & \quad \{ \text{ property 2 } \} \\
 & R \circ \tau \;\supseteq\; S \circ \Phi.R \\
\equiv & \quad \{ \text{ uep 5(b) } \} \\
 & \mathsf{true}
\end{array}$$

□

To sum up properties 8, 10 and 11:

Corollary 12 If R is a catamorphism, then $R\dagger$ is a half-congruence.
□

We now address the reverse implication: that is, we show that all half-congruences can be expressed in the form $\llparenthesis S \rrparenthesis\dagger$ for some S.

Lemma 13 $R\dagger = R$ iff R is reflexive and transitive.
Proof: By properties 8 and 9, $R\dagger$ is reflexive and transitive; thus if $R\dagger = R$, then R too is reflexive and transitive. Conversely,

$$\begin{array}{ll}
 & R \text{ is reflexive and transitive} \\
\equiv & \quad \{ \text{ definition } \} \\
 & R \supseteq \mu\Phi \quad \wedge \quad R \supseteq R \circ R \\
\Rightarrow & \quad \{ \text{ monotonicity, definition 7 } \} \\
 & R \circ R\dagger \supseteq \mu\Phi \circ R\dagger \quad \wedge \quad R\dagger \supseteq R \\
\equiv & \quad \{ R\dagger \in \mu\Phi \sim \mu\Phi \} \\
 & R \circ R\dagger \supseteq R\dagger \quad \wedge \quad R\dagger \supseteq R \\
\Rightarrow & \quad \{ \text{ property 9 } \} \\
 & R \supseteq R\dagger \quad \wedge \quad R\dagger \supseteq R \\
\equiv & \quad \{ \text{ antisymmetry } \} \\
 & R\dagger = R
\end{array}$$

□

Lemma 14 If R is reflexive and transitive, then $R = \llparenthesis R \circ \tau \rrparenthesis$ iff R is Φ-substitutive.
Proof: Let R be reflexive and transitive. We prove "if" and "only if" separately. First,

$$\begin{array}{ll}
 & R = \llparenthesis R \circ \tau \rrparenthesis \\
\equiv & \quad \{ \text{ uep 5(b) } \} \\
 & R \circ \tau = R \circ \tau \circ \Phi.R \\
\Rightarrow & \quad \{ R \supseteq \mu\Phi, \tau \in \mu\Phi \sim \Phi.\mu\Phi \} \\
 & R \circ \tau \supseteq \tau \circ \Phi.R
\end{array}$$

Now, for the other implication,

$$\begin{array}{ll}
 & R \circ \tau \supseteq \tau \circ \Phi.R \\
\Rightarrow & \quad \{ \text{ monotonicity, } R \supseteq R \circ R \} \\
 & R \circ \tau \supseteq R \circ \tau \circ \Phi.R \\
\equiv & \quad \{ \text{ see below } \} \\
 & R \circ \tau = R \circ \tau \circ \Phi.R
\end{array}$$

$\equiv \qquad \{ \text{ uep 5(b) } \}$

$\qquad R \;=\; \llparenthesis R \circ \tau \rrparenthesis$

The hint "see below" is the following:

$\qquad R \circ \tau \circ \Phi.R$

$\supseteq \qquad \{ R \supseteq \mu\Phi, \text{ relator } \Phi \text{ is monotonic} \}$

$\qquad R \circ \tau \circ \Phi.\mu\Phi$

$= \qquad \{ \tau \in \mu\Phi \leftarrow \Phi.\mu\Phi \}$

$\qquad R \circ \tau$

□

Property 15 $R = \llparenthesis R \circ \tau \rrparenthesis\dagger$ iff R is a half-congruence.

Proof:

$\qquad R$ is a half congruence

$\equiv \qquad \{ \text{ definition } \}$

$\qquad (R \text{ is reflexive}) \;\wedge\; (R \text{ is transitive}) \;\wedge\; (R \text{ is substitutive})$

$\equiv \qquad \{ \text{ 13 and 14 } \}$

$\qquad R\dagger \;=\; R \quad \wedge \quad R \;=\; \llparenthesis R \circ \tau \rrparenthesis$

$\Rightarrow \qquad \{ \text{ calculus } \}$

$\qquad R \;=\; \llparenthesis R \circ \tau \rrparenthesis\dagger$

$\Rightarrow \qquad \{ \text{ 8, 10, 11 } \}$

$\qquad (R \text{ is reflexive}) \;\wedge\; (R \text{ is transitive}) \;\wedge\; (R \text{ is substitutive})$

$\equiv \qquad \{ \text{ definition } \}$

$\qquad R$ is a half congruence

□

Now for arbitrary S, we have that $\llparenthesis S \rrparenthesis\dagger$ is a half congruence, but not necessarily symmetric, so we shall consider the intersection of such a relation with its own converse: clearly the result is reflexive, transitive and symmetric, since intersection and converse preserve reflexivity and transitivity. We need only show that intersection and converse preserve Φ-substitutivity. First converse:

Property 16 R is substitutive iff $R^{\cup}$ is.

Proof:

$\qquad R \circ \tau \;\supseteq\; \tau \circ \Phi.R$

$\equiv \qquad \{ \text{ property of } {}^{\cup}\text{, relators } \}$

$\qquad \tau^{\cup} \circ R^{\cup} \;\supseteq\; \Phi.R^{\cup} \circ \tau^{\cup}$

$\equiv \qquad \{ \text{ property 3 } \}$

$\qquad R^{\cup} \;\supseteq\; \tau \circ \Phi.R^{\cup} \circ \tau^{\cup}$

$\equiv \qquad \{ \text{ property 2 } \}$

$\qquad R^{\cup} \circ \tau \;\supseteq\; \tau \circ \Phi.R^{\cup}$

□

Property 17 Intersection preserves substitutivity.
Proof: Assume that R and S are Φ-substitutive; i.e., assume

$$R \circ \tau \supseteq \tau \circ \Phi.R \tag{18}$$

$$S \circ \tau \supseteq \tau \circ \Phi.S \tag{19}$$

Substitutivity of their intersection is proven by:

$$\begin{array}{ll}
 & (R \cap S) \circ \tau \\
= & \quad \{\ \text{set theory, } \tau \text{ is a function}\ \} \\
 & (R \circ \tau) \cap (S \circ \tau) \\
\supseteq & \quad \{\ \text{assumption; monotonicity of intersection}\ \} \\
 & (\tau \circ \Phi.R) \cap (\tau \circ \Phi.S) \\
\supseteq & \quad \{\ \text{set theory}\ \} \\
 & \tau \circ (\Phi.R \cap \Phi.S) \\
\supseteq & \quad \{\ \text{monotonicity of relators}\ \} \\
 & \tau \circ \Phi.(R \cap S)
\end{array}$$

□

Properties 16 and 17 together show that substitutivity of $\llparenthesis S \rrparenthesis\dagger \cap \llparenthesis S \rrparenthesis\dagger^{\cup}$ follows from substitutivity of $\llparenthesis S \rrparenthesis\dagger$, which has already been established. In all we have proven:

Theorem 20 Relation $R \in \mu\Phi \sim \mu\Phi$ is a congruence relation iff $R = \llparenthesis S \rrparenthesis\dagger \cap \llparenthesis S \rrparenthesis\dagger^{\cup}$ for some S.
□

3 Induced Congruence Relations

We conclude with a proof of a theorem on congruence relations which are induced by functional catamorphisms. A weaker version of the theorem (an implication rather than an equivalence) occurs as a standard exercise in many universal algebra textbooks; the stronger version given here is stated (with a proof only of the implication) in Ehrig & Mahr ([4], p.77).

Theorem 21 For (total) function $f \in \alpha \leftarrow \mu\Phi$, f is a catamorphism iff $f^{\cup} \circ f$ is a congruence relation.

Proof: To see that if f is a catamorphism then $f^{\cup} \circ f$ is a congruence, note first that $f^{\cup} \circ f$ is equal to $f\dagger$, since for all X

$$\begin{array}{ll}
 & f\dagger \supseteq X \\
\equiv & \quad \{\ \text{definition of } \dagger\ \} \\
 & f \supseteq f \circ X \\
\equiv & \quad \{\ \text{property 3}\ \} \\
 & f^{\cup} \circ f \supseteq X
\end{array}$$

Moreover, $f^{\cup} \circ f$ is obviously symmetric, so we have $f^{\cup} \circ f = f\dagger \cap f\dagger^{\cup}$. It follows by theorem 20 that if f is a catamorphism, then $f^{\cup} \circ f$ is a congruence relation.

The other implication is proven as follows:

$$
\begin{array}{ll}
 & f^{\cup} \circ f \ \text{ is a congruence} \\
\Rightarrow & \quad \{ \text{ lemma 14 } \} \\
 & f^{\cup} \circ f \ = \ (\!|f^{\cup} \circ f \circ \tau|\!) \\
\equiv & \quad \{ \text{ uep 5(b) } \} \\
 & f^{\cup} \circ f \circ \tau \ = \ f^{\cup} \circ f \circ \tau \circ \Phi.(f^{\cup} \circ f) \\
\Rightarrow & \quad \{ \text{ monotonicity } \} \\
 & f \circ f^{\cup} \circ f \circ \tau \ = \ f \circ f^{\cup} \circ f \circ \tau \circ \Phi.(f^{\cup} \circ f) \\
\equiv & \quad \{ \ f = f \circ f^{\cup} \circ f, \text{ definition of a relator 4(c) } \} \\
 & f \circ \tau \ = \ f \circ \tau \circ \Phi.f^{\cup} \circ \Phi.f \\
\equiv & \quad \{ \text{ uep 5(b) } \} \\
 & f \ = \ (\!|f \circ \tau \circ \Phi.f^{\cup}|\!)
\end{array}
$$

□

4 Conclusion

In this paper we have given a complete characterisation of a congruence relation on an initial algebra in terms of relational catamorphisms. Whether or not this will prove to be a particularly useful characterisation we cannot yet say. However, we are strongly encouraged by the economy of our calculations which is a major driving force in our work. The economy that has been achieved here can be traced back to the concise statment of the initiality of $\mu\Phi$ and the use of three Galois correspondences, namely, properties 2 and 3 and definition 7.

Acknowledgement

All proofs in this document were prepared using the proof editor system developed by Paul Chisholm [3].

References

[1] R.C. Backhouse. Naturality of homomorphisms. Lecture notes, International Summer School on Constructive Algorithmics, vol. 3, 1989.

[2] R.C. Backhouse, P. de Bruin, G. Malcolm, E. Voermans, and J. van der Woude. Types and relations. Eindhoven University of Technology and University of Groningen, September 1990.

[3] P. Chisholm. Calculation by computer: Overview. Technical Report CS 9007, Department of Computing Science, University of Groningen, 1990.

[4] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 1*. EATCS Monographs on Theoretical Computer Science. Springer-Verlag Berlin, 1985.

[5] J.A. Goguen, J.W. Thatcher, and E.G. Wagner. An initial algebra approach to the specification, correctness and implementation of abstract data types. In R.T. Yeh, editor, *Current Trends in Programming Methodology, Volume 4: Data Structuring*, pages 80–149. Prentice-Hall, 1978.

[6] T. Hagino. A typed lambda calculus with categorical type constructors. In D.H. Pitt, A. Poigne, and D.E. Rydeheard, editors, *Category Theory and Computer Science*, pages 140–57. Springer-Verlag Lecture Notes in Computer Science 283, 1988.

[7] C.A.R. Hoare and Jifeng He. The weakest prespecification. *Fundamenta Informaticae*, 9:51–84, 217–252, 1986.

[8] G. Malcolm. Data structures and program transformation. To appear, *Science of Computer Programming*, 1990.

[9] G. Malcolm. *Algebraic data types and program transformation.* PhD thesis, Groningen University, 1990.

[10] E.G. Manes and M.A. Arbib. *Algebraic Approaches to Program Semantics.* Texts and Monographs in Computer Science. Springer-Verlag, Berlin, 1986.

[11] J. Meseguer and J.A. Goguen. Initiality, induction and computability. In M. Nivat and J.C. Reynolds, editors, *Algebraic Methods in Semantics*, pages 459–542. Cambridge University Press, 1985.