TM-1057

Unification in Final Coalgebras
— Extended Abstract —

by
K. Mukai

June, 1991

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191 ~ 5
Telex ICOT J32964

Institute for New Generation Computer Technology

Unification in Final Coalgebras*
——Extended Abstract——

Kuniaki Mukai

Institute for New Generation Computer Technology (ICOT)
Mita Kokusai Bldg. 21F, 1-4-28 Mita, Minato-ku, Tokyo 108 JAPAN
Tel: +81-3-456-3069, E-mail: mukai@icot.or.jp

1 Introduction

The objective of this paper is to apply a final coalgebra theorem given in Aczel[1] and Aczel and Mendler[2] on the category of classes of hypersets to several standard structured objects such as infinite and rational trees[5], finite automata, feature (=record) structures. Barwise[3] gives a unification theorem for the mixed constraints consisting of bisimulations and subsumptions on the class V of hypersets with urelements. His proof is based on Aczel's Solution Lemma[1]. V is the final coalgebra for *pow* the class functor and Solution Lemma is a special case of the final coalgebra theorem, though the former is a foundation of the whole hyperset theory. This paper generalizes the unification theorem from the one based on the solution lemma to the one based on the final coalgebra theorem. By doing so, we can treat uniformly infinite tree unification and feature unification as an instance of Barwise's unification theory, which was implicit in Barwise[3].

The generalized unification theorem holds for a large class of set-based functors. In fact we prove the theorem for a class of natural functors called *pure* and *subterm-closed*. Thus Barwise' unification theorem for the power class functor is generalized for the functors which are set-based, pure, and subterm closed.

In particular, we treats in this paper the following functor pow, pow', H^A, $map(A, -)$, where A is a class of urelements as a signature, and

- $H^A(M)$ is the class of terms t such that prime function symbols of t is in A and arguments of t are in M.
- $pow'(M)$ is the class of finite sets of M.
- $map(D, D')$ is the class of partial maps from D into D'.

H^A is used as a Herbrand universe forming operator. $map(A, -)$ is used as a common forming operator for domains of records, finite automata, rational trees, with a slight modification

*Under submission to 3rd STA conference, Japan, 1991.

for each class of domains. As a (deterministic) automata is simply a coalgebra for $map(A, -)$, from this view based on the final coalgebra theorem, a relationship is given between automata, rational trees, and regular languages.

A kind of record merge operations is defined as a coalgebra for $map(A, -)$. The class of extensional subsumption constraint satisfaction problems is solved through this coalgebra. This problem was first solved recently in Dörre[6] by reducing to a well-known standard method in finite automata theory. As automata can not be an initial algebra in general, this result shows an advantage for final coalgebra to initial algebras.

In this paper constraints are formalized as binary relations on *initial* algebras of these class functors. In particular, the notion of solutions in final coalgebra are defined through the Aczel's final coalgebra theorem.

Although details remains to be written down, the results of this paper seem to be straightforward consequences of the final coalgebra theorem. I am writing this report with a surprise seeing that the final coalgebra theorem gives an integrated and unified view on those familiar structured objects. This unified view seems to have been long implicit so far.

Basic tools of this paper is the final coalgebra theorem[1, 2] and the unification theorem in Barwise[3]. This work is a continuation of Mukai[8], which applies the solution lemma to the constraint logic programming scheme. Barwise and Etchemendy[4] gives an introductory description to the final coalgebra theorem as a part of introduction of the hyperset theory.

It seems to be very interesting to use Hagino's categorical programming language CPL[7] for implementing the final coalgebra theorem. Roughly speaking, CPL works on a cartesian closed category. Initial and final objects are introduced by dialgebras, that are a generalization of a T-algebra. Note that the final coalgebra theorem is formalized in a language of T-algebra. Thus, this work will continue to have a foundation for a new programming language for both initial (left, well-founded) or final (right, non-well-founded) structured objects. This kind of research in the near future seems to be strongly necessary for implementing situation theory because of complex structured objects.

2 Basic Definitions

We assume a universe V of hypersets with urelements for our metatheory. Δ denotes the class of urelements in V: $\Delta \subseteq V$. Urelements will be called *atoms* or *parameters* depending on the context. Elements of V that are not urelements are called *sets*. The emptyset $\emptyset$ is a set but not an urelement. If x is an urelement then there is no $y \in V$ such that $y \in x$. A *class functor* is an endo-functor on the superlarge category of classes on V. Let T be a class functor. $I(T)$ denotes the minimum fixpoint of T. $J(T)$ denotes the maximum fixpoint of T. If T is set-based then $I(T)$ and $J(T)$ exist[1]. By T_C we mean the functor defined from T by $T_C(M) = C \cup T(M)$. The notation T_C is fundamental in this paper. Then V is the maximum fixpoint of a functor pow_Δ, where pow is the power class functor, i.e., $pow(M)$ is the class of subsets of M. A *n-ary relation* is a subclass of V^n. The *carrier* of a n-ary relation r is the class $\{x_i \in V \mid r(x_1, \cdots, x_i, \cdots, x_n), 1 \leq i \leq n\}$. $carri(r)$ denotes the carrier of r. A class t is called *transitive* if $t \subseteq pow(t)$. Given a class $u \in V$, $trans(u)$ denotes the minimum transitive class $t \in V$ such that $u \subseteq t$. $trans(u)$ is called the *transitive closure* of u. For a relation r on V, we define $fld(r) = trans(carri(r))$.

Example 1

$$fld(\{(x, a), (y, \{b, c\})\}) = \{x, a, y, \{b, c\}, b, c\}$$

where x, y, a, b, and c are not set terms. □

$\mathcal{V}_X(s) \stackrel{\text{def}}{=} X \cap trans(\{s\})$, where $s \in V$, $X \subseteq \Delta$. A pow' is the functor such that $pow'(M)$ is the class of finite subsets of M.

Elements of V are called *terms*. Sets in $J(pow'_\Delta)$ are called *hereditarily finite*. If A is a set, $J(pow'_A)$ forms a set. For any $x \in I(pow'_\Delta)$, $trans(x)$ is finite. So we call terms in $I(pow'_\Delta)$ *finitary*. In fact, finitary set terms are hereditarily finite and well-founded.

Example 2 Let N be the set of natural numbers. Then the set $\{N\}$ is finite and wellfounded, but not finitary since $trans(\{N\})$ is infinite. □

Definition 1 Let $A \subseteq \Delta$, and assume that each $a \in A$ is assigned a non-negative integer. Then H^A is defined by $H^A(M) = \{(a, \sigma) \mid a \in A, \sigma\colon \{1, \cdots, n_a\} \to M\}$ with n_a being the arity of a, where $M \subseteq V$. □

Elements of $J(H^A_X)$ and, in particular, $J(H^A)$ are called *Herbrand terms* and *Herbrand trees*, respectively. In this translation we have identified each atom a of arity 0 with $(a, \emptyset)$.

It is clear that $J(H^A_X)$ is a proper subset of $J(pow'_{A \cup X})$. There is a natural bijection ψ from the the standard first-order term notations onto the finitary Herbrand terms such that $\psi(f(x_1, \ldots, x_n)) = (f, \{(i, \psi(x_i)) \mid 1 \leq i \leq n\})$.

Definition 2 Let $X \subseteq \Delta$ be a class. A family $(b_x)_{x \in X}$ is called a *system of equations* (for X) if $b_x \in V \setminus X$ for each $x \in X$. □

Theorem 1 (Solution Lemma [1]) *Every system of equations for a class X has a unique solution in $J(pow_{\Delta \setminus X})$.*

Definition 3 ([2]) An endo-functor T on the superlarge category of classes is called *set-based* if for each class A and each $a \in T(A)$ there is a set $A_0 \subset A$ and $a_0 \in T(A_0)$ such that $a = T_{\iota_{A_0,A}}(a_0)$, where $\iota_{A_0,A}$ is the inclusion map $A_0 \hookrightarrow A$. □

Theorem 2 (Final Coalgebra Theorem [1, 2]) *Every set-based functor has a final coalgebra.*

Note that the solution lemma is a special case of the final coalgebra theorem.

3 Constraints

Definition 4 A *constraint (on D)* is an ordered pair $\langle r, X \rangle$, where r is a binary relation on $D \subseteq V$ and $X \subseteq \Delta$. □

We often write r for $\langle r, X \rangle$ with X being implicit and call it an *X-constraint*. A constraint $\langle r, X \rangle$ is an *extension* of a constraint $\langle s, Y \rangle$ if $s \subseteq r$, $Y \subseteq X$, $\mathcal{V}_X(r) = \mathcal{V}_Y(s)$. An *assignment* is a partial function from Δ into V. Let f be an assignment. Then it follows from the solution lemma that there is a function $\hat{f}$ from V to $J(pow_{\Delta \setminus dom(f)})$ such that

(1) $\hat{f}(x) = f(x)$ if $x \in dom(f)$.

(2) $\hat{f}(a) = a$ if $a \in A \setminus dom(f)$.

(3) $\hat{f}(s) = \{\hat{f}(u) \mid u \in s\}$ if s is a set.

An function f is called an *assignment for* a constraint $\langle r, X\rangle$ if $dom(f) = X$ and $ran(f) \subseteq J(pow_{\Delta\setminus X})$.

Definition 5 An assignment f for a constraint c is called a *solution* of c in $D \subseteq V$ if $ran(f) \subseteq D$ and $\hat{f}(u) = \hat{f}(v)$ whenever $c(u,v)$. □

Definition 6 Let $X \subseteq \Delta$. A *bisimulation* is a constraint $\langle r, X\rangle$, where r is an equivalence relation on some $D \subseteq V$ such that if $r(u,v)$ then the following hold.

- If $u \in \Delta \setminus X$ and $v \notin X$ then $u = v$.
- If u and v are sets then

$$\forall x \in u \exists y \in v\ r(x,y) \quad \& \quad \forall y \in v \exists x \in u\ r(x,y).$$

□

Definition 7 Let $X \subseteq \Delta$. $x \in \Delta$ is *bound* in a constraint $\langle c, X\rangle$ if $x \in X$ and $c(x,b)$ for some $b \notin X$. A constraint $\langle c, X\rangle$ is *bound* if each $x \in X$ is bound in $\langle c, X\rangle$. □

Clearly it follows from the solution lemma that a bound constraint $\langle r, X\rangle$ has at most one solution. We say a constraint $\langle s, Y\rangle$ is a *bisimulation of* a constraint $\langle r, X\rangle$ if $\langle s, Y\rangle$ is a bisimulation and an extension of $\langle r, X\rangle$. Also we say a constraint r *has* a bisimulation s when s is a bisimulation of r. We say that a constraint $\langle s, X\rangle$ is a *small bisimulation* of a constraint $\langle r, Y\rangle$ if $X = Y$ and $\langle s, Y\rangle$ is a bisimulation of $\langle r, X\rangle$ and $fld(r) \subseteq fld(s)$, i.e., only terms appearing in s appear in r. Constraints c are called *finitary* if c is finitary as terms.

Proposition 3 *For every finitary constraint c, the following hold.*

(1) The set of small bisimulations of c is finite.

(2) If there exists a bisimulation of c, then there exists also a small bisimulation of c.

Proof Let $Q = \{(x,y) \mid trans(\{x,y\}) \subseteq fld(c)\}$. As c is finitary, $fld(c)$ is finite. So Q must be finite. As every small bisimulation of c is a subset of Q, we get (1).

We prove (2). Let p be a bisimulation of c. By definition, we get $c \subseteq p$. As Q is finite, $p \cap Q$ is finite. It is a routine to check that the constraint $p \cap Q$ satisfies all defining clauses of a bisimulation. As $c \subseteq Q$, we get $c \subseteq p \cap Q$. Therefore $p \cap Q$ is a small bisimulation of c. □

Proposition 4 *For every finitary constraint c, the existence of a bisimulation of c is decidable.*

Proof In general, for a given finite set B, the existence of a bisimulation p such that $fld(p) \subseteq B$ is decidable by an exhaustive search method.

Clearly, for all constraints p it follows that p is a small bisimulation of c iff p is a bisimulation and $fld(p) \subseteq fld(c)$. Since c is finitary, $fld(c)$ is finite. So it follows from the above general remark that the existence of a small bisimulation of c is decidable. Hence, by Proposition 3, the existence of a bisimulation of c is decidable. □

Definition 8 A functor T is called *pure* if, for all classes $X \subseteq \Delta$ and $M \subseteq V$, $X \cap trans(M) = \emptyset$ implies $X \cap trans(T(M)) = \emptyset$. pow is pure. □

pow_A is pure, where $A \subsetneq \Delta$. The unification theorem is a straightforward generalization of Barwise's unification theorem[3].

Theorem 5 (Unification Theorem) *Let T be a pure and set-based functor, and $X \subseteq \Delta$. Then for every constraint $\langle p, X\rangle$ on $I(T_X)$ with $X \cap trans(J(T)) = \emptyset$, the following are equivalent.*

(1) $\langle p, X\rangle$ is solvable in $J(T)$.

(2) $\langle p, X\rangle$ extends to some bound constraint $\langle q, Y\rangle$ on $I(T_Y)$ that has a bisimulation.

Proof (2)$\Longrightarrow$(1): Suppose (2) is true. Let $\langle q, X\rangle$ be a bound constraint on $J(T_X)$ that is an extension of $\langle p, X\rangle$ and has a bisimulation $\langle r, Y\rangle$. We assume, without loss of generality, that $\langle r, Y\rangle$ is bound. Define a family $(b_y)_{y \in Y}$ of terms so that $q(x, b_x)$ and $b_x \in I(T_X) \setminus X$ for each $x \in X$. By the final coalgebra theorem[1, 2], there is a solution f of $y = b_y$ for all $y \in Y$ in $J(T)$. Let $r = \{(\hat{f}(u), \hat{f}(v)) \mid q(u, v)\}$. Clearly $r \subseteq J(T) \times J(T)$. Moreover, as r is a bound bisimulation, it follows that r is a $\emptyset$-bisimulation relation on $J(T)$. As it follows from [1] that $\emptyset$-bisimulation on $J(pow_{\Delta\setminus Y})$ is an identical relation[1], we get $r \subseteq =$. Hence we get $\hat{f}(u) = \hat{f}(v)$ for all u, v such that $q(u, v)$. So f is a solution of q. As $p \subseteq q$, f is a solution of p.

(1)$\Longrightarrow$(2): Suppose (1) is true. Let f be a solution of p. For each $x \in X$, let $E_x = (b^x_z)_{z \in K_x}$ be a system of equations such that $b^x_z \in I(T_{K_x})$ for all $z \in K_x$, where $K_x \subseteq \Delta$, $K_x \cap X = \{x\}$, $K_y \cap K_z = \emptyset$ for $z \neq y$, and $f(x) = \widehat{\tau_x}(b^x_x) \in J(T)$, where τ_x a solution of E_x for $x \in X$. Define $q = p \cup \{(z, b^x_z) \mid x \in X, z \in K_x\}$, and $Y = \bigcup\{K_x \mid x \in X\}$. Then clearly $\langle q, Y\rangle$ is a bound constraint on $J(T_Y)$ and an extension of $\langle p, X\rangle$. Moreover, as $\langle q, Y\rangle$ has a solution, say g, q has a bisimulation $\langle\{(u, v) \mid u, v \in fld(q), \hat{g}(u) = \hat{g}(v)\}, Y\rangle$. Therefore we get (2). □

Definition 9 A class functor T is *subterm-closed* if the following are equivalent for any $X \subseteq \Delta$ and X-constraint p on $I(T_\Delta)$.

(1) p has a small X-bisimulation, i.e., one on $fld(p)$.

(2) p has a small X-bisimulation q such that for each $x, y \in X$, $q(x, y)$ implies $y \in I(T_\Delta)$. □

Example 3 H^A, pow, pow' and Π are subterm-closed, where Π is a record constructing functor described below. □

Definition 10 Given a functor T, a T-unification problem is to decide whether given constraints $\langle c, X\rangle$ on $I(T_\Delta)$ are solvable in $J(T_{\Delta\setminus X})$. □

Let q be a bisimulation extension q of an X-constraint p on $I(T_X)$. Then $q \cap (fld(p) \times fld(p))$ is an X bisimulation on $fld(p)$ that is an extension of p. So, if X and $fld(p)$ are finite, we can enumerate all X-bisimulation extensions of p on $fld(p)$. So we have the theorem.

Theorem 6 *If T be subterm-closed, pure, and set-based then any finitary T-unification problem are decidable.*

Hence, when T is subterm-closed, p is solvable iff p extends to a X-bisimulation r on $fld(p)$. Therefore the solvability of p in $J(T)$ is decidable provided that X and $fld(p)$ are finite.

Colmerauer's unification theory[5] on infinite trees (without unequations) falls into this class with $T = H^A$. Furthermore, in fact, the infinite tree unification with unequations can be treated without much difficulty by a straightforward modification of the notion of a constraint and a solution. However it is outside of the scope of the paper and will appear elsewhere.

4 Functors H^A, pow', Π

Recall the functor H^A. We identify $J(H^A)$ with the domain of infinite trees in Colmerauer[5] over A. Let $A \subseteq \Delta$ and let R_A be the set of $u \in J(pow'_A)$ that has a finite transitive closures. Elements of R_A are called *rational sets*. Then, for any given finite constraint c on $I(pow'_A)$, we can know by an effective method whether c is solvable in R_A or not by. The functor H^A is subterm-closed. So we have the theorem.

Theorem 7 *For any X-constraint c on $J(H^A_X)$, the following are equivalent:*

(1) c is solvable in $J(H^A)$.

(2) c is solvable in $J(pow'_A)$.

Let $F, A \subseteq \Delta$ be disjoint two sets. Elements F are called *features*. A special atom $\bot \in \Delta \setminus (F \cup A)$ means an undefined values of algebras. Define a class functor

$$\Pi = map(F, -).$$

Clearly Π_B is pure, set-based and subterm-closed for any $B \subseteq \Delta$. Elements of $J(\Pi_A)$ are called *records* over (F, A).

We define a merge operation on records as a coalgebra $(pow(J(\Pi_A)), \mu)$ for $\Pi_{\{\bot\}\cup A}$ by

- $\mu(u) = \bot$ if there are $x, y \in u$ such that $x \neq y$ and either $x \in A$ or $y \in A$.
- $\mu(u) = a$ if $a \in A \cap u$.
- $\mu(u) = f_u$ if u is a set, where $dom(f_u) = \bigcup\{dom(g) \mid g \in u\}$, $f_u(\nu) = \{f(\nu) \mid f \in u, \nu \in dom(f)\}$.

μ is well-defined. By the final coalgebra theorem, there is a function $\pi: pow(J(\Pi_A)) \to J(\Pi_{A\cup\{\bot\}})$ such that $\pi(u) = \Pi_{\{\bot\}\cup A}(\pi)(\mu(u))$, for $u \in pow(J(\Pi_{\{\bot\}\cup A}))$. The operation μ is a *record merge operations*. Define $u^* = \Pi_{\{\bot\}\cup A}(\pi)(\mu(u))$.

Example 4

$$\{\{(\nu, a)\}, \{(\nu, b)\}\}^* = \bot, \text{ where } a \neq b \in A.$$
$$\{\{(\nu_1, a)\}, \{(\nu_2, b)\}\}^* = \{(\nu_1, a), (\nu_2, b)\}, \text{ where } \nu_1 \neq \nu_2 \in F.$$

□

We use $\sqsubseteq_A$ for the maximum $\emptyset$-subsumption relation on $J(\Pi_A)$. The notion of a solution of X-subsumption constraint on $I(\Pi_{A\cup X})$ in $J(\Pi_A)$ is defined in a similar way to the bisimulation constraint.

Proposition 8 *The following are equivalent.*

(1) For all $w \in v$, $w \sqsubseteq u$.

(2) $u^* \sqsubseteq v$.

The following definition is a slight modification of Barwise [3].

Definition 11 Let $R = J(\Pi_{A\cup X})$ for disjoint $A, X \subseteq \Delta \setminus F$. A constraint (p, X) is called a *(X-)subsumption* on R if p is reflexive and symmetric binary relation on R satisfying the following.

(1) If $x \in A$, $y \notin X$, $p(x, y)$ then $x = y$.

(2) If $x \in A$, $y \notin X$, $p(y, x)$ then $x = y$.

(3) If $x, y \notin \Delta$, $p(x, y)$ then $dom(x) \subseteq dom(y)$ and $p(x(\nu), y(\nu))$ for all $\nu \in dom(x)$.

□

We define an binary operation on records for merging *parametric* records. Let $b = (b_x)_{x \in X}$ be a system of equations such that $b_x \in I(\Pi_{A \cup X}) \setminus X$. Then:

Definition 12 $\mu_b(u) \stackrel{\text{def}}{=} \mu(u')$, where $u' = \{f \in u \mid f \in u \setminus X\} \cup \{b_x \mid x \in u \cap X\}$. □

Definition 13 Given a X-bisimulation p and X-subsumption on $I(\Pi_{A \cup X})$. A *record compatibility relation* r is a binary relation on $I(\Pi_{A \cup X})$ defined w.r.t. p, q satisfying the following.

(1) $p \subseteq r$.

(2) If for some z, $q(x, z)$ and $q(y, z)$, then $r(x, y)$.

(3) If $r(x, y)$, then for all $\nu \in dom(x) \cap dom(y)$, $r(x(\nu), y(\nu))$.

□

Given p, q, r as the above, for each unbound $x \in X$ in p, define $R_x = \{y \mid r(x, y), y \in X\}$. By the final coalgebra theorem, $\pi(x) \in J(\Pi_{\{\perp\} \cup A})$. Moreover, as each equivalence class of r has no *conflict*, by the final algebra theorem, $\perp \notin trans(\pi(x))$. This method contains decision procedure for subsumption problem, which recently Dörre[6] first solved by using a well-known method for transforming non-deterministic finite automata into deterministic ones. Our work gives an account to the solution from the final coalgebra theorem[1, 2].

4.1 Finite Automata

Automata are coalgebras for a class functor $\Sigma = map(A, -) \times pow(\{\lambda\})$, where $A \subseteq \Delta$ and $\lambda \in \Delta$ is a distinguished urelement for indicating 'accept' states of the automata, and also the empty string of regular language. Equivalently,

$$\Sigma(M) = \{u \cup v \mid u \subseteq \{\lambda\}, u \in map(A, M)\}.$$

Those sets in $J(\Sigma)$ that have finite transitive closures are called *rational trees*. Rational trees are sets of the form $r = u \cup v$, where $u \subseteq \{\lambda\}$, $v \in map(A, J(\Sigma))$. If r is a rational trees then $L(r)$ is a regular language over A, where $L(r) = L'(u) \cup L''(v)$, $L'(x) = x$, $L''(u) = \bigcup\{aL(u(a)) \mid a \in dom(u)\}$.

Let (Y, δ) be a finite coalgebra for Σ and π is a mediating arrow to the final coalgebra of Σ. Then $\pi(x)$ is a rational trees for all $x \in Y$. These results are obtained straightforwardly without much difficulty.

References

[1] P. Aczel. *Non-well-founded Sets*. CSLI lecture notes Number 14. CSLI Publications, Stanford University, 1988.

[2] P. Aczel and N. Mendler. A final colagebra theorem. In P.H. Pitt, D.E. Rydcheard, P. Dybjer, A.M. Pitts, and A. Poigné, editors, *Category Theory and Computer Science*, number 389 in LNCS. Springer-Verlag, 1989.

[3] J. Barwise. *The Situation in Logic.* CSLI Lecture Notes 17. Standord: CSLI Publications, 1989.

[4] J. Barwise and J. Etchemendy. *The Liar: An Essay on Truth and Circular Propositions.* Oxford Univ. Press, 1987.

[5] A. Colmerauer. Equations and unequations on finite and infinite trees. In *Proceedings of the Second International Conference on Fifth Generation Computer Systems*, Tokyo, 1984.

[6] J. Dörre. Feature logic with weak subsumption constraints. Technical Report IWBS tech. Report 101, IBM Deutschland,Stuttgart, April 1990.

[7] T. Hagino. *A Categorical Programming Language.* PhD thesis, University of Edinburgh, 1987.

[8] K. Mukai. CLP(AFA): Coinductive semantics of horn clauses with compact constraints. In J. Barwise, G. Plotkin, and J.M. Gawron, editors, *Situation Theory and Its Applications, volume II.* CSLI Publications, Stanford University, 1991. to appear.