

A Calculus of Terms for Coalgebras of Polynomial Functors

Robert Goldblatt¹

*Centre for Logic, Language and Computation
School of Mathematical and Computing Sciences
Victoria University of Wellington, New Zealand*

Rob.Goldblatt@vuw.ac.nz <http://www.mcs.vuw.ac.nz/~rob>

Abstract

A syntax and semantics of types, terms and formulas for coalgebras of polynomial functors is developed, extending earlier work [4] on monomial coalgebras to include functors constructed using *coproducts*. A modified ultrapower construction for polynomial coalgebras is introduced, adapting the conventional ultrapower to retain only those states that evaluate observable terms in a standard way.

A special role is played by terms that take observable values and are “rigid”: their free variables do not occur in any state-valued subterm. The following “co-Birkhoff” theorem is proved: a class of polynomial coalgebras is definable by Boolean combinations of equations between rigid terms iff the class is closed under disjoint unions, images of bisimulations, and observable ultrapowers.

1 Introduction

A coalgebra of a functor $T : \mathbf{Set} \rightarrow \mathbf{Set}$ is a pair (A, α) with α a function of the form $A \rightarrow TA$. This notion has proven useful in modelling transition systems, such as automata, as well as classes in object-oriented programming languages [14,7,16,17]. α is viewed as a *transition structure* on a *state set* A .

Relational models of propositional modal logic can be viewed as coalgebras [16] and this has lead to a number of proposals of languages with modalities for describing coalgebras [12,11,15,10]. An alternative method used here is to develop a syntax of equations between terms for coalgebraic operations that is similar to the standard equational logic of algebras, but subject to the principle that a coalgebraic term should have a single *state*-valued variable or parameter.

¹ Paul Taylor’s `diagrams` package was used in preparing this document.

In a previous article [4] the author developed such a calculus of terms and equations for coalgebras of certain *monomial* functors. These are constructed from constant functors and the identity functor by forming products and exponential functors with constant exponent (which we will call *power* functors). It was shown that Boolean combinations of equations between terms of “observable” type form a suitable language of formulas for specifying properties of coalgebras and characterising bisimulation relations between them. A structural description was given of classes of coalgebras definable by such formulas, using the notion of the *ultrafilter enlargement* of a monomial coalgebra.

Now many of the more significant examples in the above references involve also *coproducts* in their construction, and so are coalgebras for *polynomial* functors. The aim of this article is to explain how the theory of [4] can be extended to the case of polynomial functors. The presence of coproducts introduces considerable complexity, associated with the *partiality* of certain “path functions” that express the dynamics of the transition structure α .

The approach taken here is to use type theory [8] to describe the construction of sets-as-types from some base types by forming products, powers and coproducts, and to provide rules of syntax for terms that take values in these types. Among the base types is the type **St** of states: this symbol **St** denotes the state set of a given coalgebra. The symbol **s** is reserved as the special state-valued parameter that appears in terms, and may be thought of as denoting the “current” state. The symbol **tr** denotes the transition structure, so that we are able to form the term $\text{tr}(\mathbf{s})$, or more generally $\text{tr}(M)$ for any state-valued term M . But the situation is far more subtle than previously, because we now allow state variables distinct from **s** in coalgebraic specifications, provided that they are not free. In the syntax of [4] all variables of a term are free, but here we have variable-binding operations on terms (lambda-abstraction, *case*-formation). A given term M may contain free state variables. More generally it may have a number of free variables of various types that occur in state-valued subterms, and hence provide a number of ways of referring to states by varying the values of those variables. M is *rigid* if this does not hold, i.e. if any variable occurring in a state-valued subterm is bound in M itself (an example will be given shortly). Rigidity is imposed on M by requiring that the type of any free variable of M does not involve **St**. Our main result (Theorem 7.1) is about the specification of coalgebras by combinations of equations between rigid terms.

Following established practice in categorical logic, the “case” operation is used to introduce terms associated with coproducts. The coproduct $A_1 + A_2$ of sets A_1, A_2 is their disjoint union, and comes equipped with injective *insertion* functions $\iota_j : A_j \rightarrow A_1 + A_2$ for $j = 1, 2$. Each element of $A_1 + A_2$ is equal to $\iota_j(a)$ for a unique j and a unique $a \in A_j$. Our syntax generates terms of the form

$$\text{case } N \text{ of } [\iota_1 v_1 \mapsto M_1 \mid \iota_2 v_2 \mapsto M_2],$$

where N is a term taking values in $A_1 + A_2$, M_1 and M_2 take values in some

other set B , and the v_j 's are variables that take values in A_j and are bound in the overall **case** term. The latter is evaluated by first obtaining the value d of N in $A_1 + A_2$ and then, if d is equal to $\iota_j(a)$, evaluating M_j with v_j assigned value a . Another notation for this term [8, Section 2.3] is

$$\text{unpack } N \text{ as } [\iota_1 v_1 \text{ in } M_1, \iota_2 v_2 \text{ in } M_2].$$

Example. To illustrate the use of rigid terms and **case**-formation in coalgebraic specification, here is an example adapted from [9, Section 4]. Let A be a set of (possibly infinite) binary trees. Each tree x either is a single node with no children, or has exactly two children obtained by deleting the top node of x . This gives an operation

$$\text{children} : A \longrightarrow 1 + (A \times A),$$

where $1 = \{*\}$; $\text{children}(x) = \iota_1*$ when x has no children, and $\text{children}(x) = \iota_2(x_1, x_2)$ when x_1 and x_2 are the left and right children of x . There is a size (number of nodes) operation

$$\text{size} : A \longrightarrow 1 + \mathbb{N},$$

where \mathbb{N} is the set of positive integers and $\text{size}(x) = \iota_1*$ when x is infinite. The two operations can be “tupled” into a single function

$$A \xrightarrow{\alpha} (1 + (A \times A)) \times (1 + \mathbb{N})$$

which is a coalgebra for the functor $T(X) = (1 + (X \times X)) \times (1 + \mathbb{N})$. The operations can be recovered from α as $\text{children} = \pi_1 \circ \alpha$ and $\text{size} = \pi_2 \circ \alpha$, where π_1 and π_2 are the left and right projections.

Now the size of a tree is 1 if it has no children, is infinite if at least one child is infinite, and otherwise is the sum of the sizes of the children plus 1. Thus our example *validates* the equation of Figure 1, in which the right-hand term M is obtained by iteration of **case**-formation. Validity means that the equation is satisfied no matter what member of A is denoted by the state parameter s . The variable v takes values in $A \times A$, so $\pi_1 v$ and $\pi_2 v$ take values in A . Although v is free in these subterms, and indeed in the subterms beginning **case** $\text{size}(\pi_j v) \dots$, v is bound in M itself. M is rigid.

A significant departure from [4] is to replace the notion of ultrafilter enlargement by a modified ultrapower. There is an obstacle to using the conventional ultrapower construction in that it produces states that assign “non-standard” values to terms of observable type. Our modification is to retain only those states that are *observable* in the sense that they assign only standard observable values (see Section 6). One advantage of ultrapowers over ultrafilter enlargements is that lifting the operations of a coalgebra to an ultrapower is a more familiar exercise, and is less cumbersome in that it works

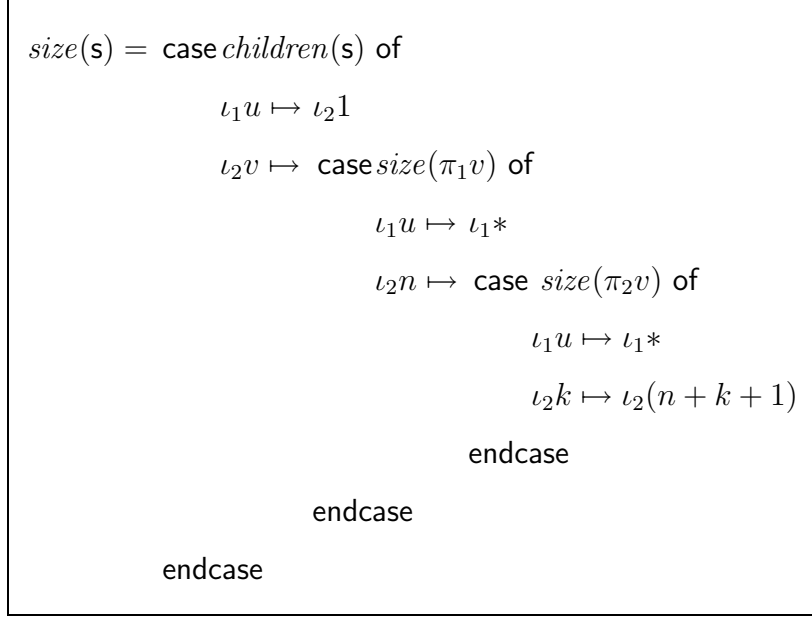


Fig. 1. case terms

with elements rather than collections of sets. Also the proof that ultrapowers preserve satisfaction of observable formulas is more accessible, and follows the pattern of Łoś's Theorem for regular ultrapowers. On the other hand there is considerable intricacy in defining the transition structure of an observable ultrapower. This is carried out with the help of the notion from [10] of a *path* from a functor to one of its component functors.

This article is in the nature of a research announcement, giving a survey of all the relevant concepts and explaining the results, but leaving out the more technical proofs, which would take up much more space than is available here (these proofs will appear elsewhere). To summarize, the main features of the work are:

- The formulation of syntax and semantics of types and terms for coalgebras of any polynomial functor (Sections 3 and 4).
- The definition of *observable* formulas as Boolean combinations of equations between terms of observable type, and their use in logically characterising bisimilarity of states: two states are bisimilar when they assign the same values to all ground observable terms, or equivalently when they satisfy the same rigid observable formulas (Theorem 5.8).
- The construction of observable ultrapowers of polynomial coalgebras and derivation of a version of Łoś's Theorem (Section 6).
- A proof that a class of polynomial coalgebras is definable by a set of observable formulas if, and only if, it is closed under disjoint unions, images of bisimulations, and observable ultrapowers (Theorem 7.1).

This last result may be viewed as an analogue for polynomial coalgebras of

Birkhoff's famous characterisation of varieties of classical algebras. For discussion of the nature of such "co-Birkhoff" theorems and references to other proposals for them, see the Introduction to [4].

2 Polynomial Functors

Standard notation for products, powers and coproducts of sets will be used. The *coproduct* $A_1 + A_2$ and associated *insertions* $\iota_j : A_j \rightarrow A_1 + A_2$ have already been described. $\pi_j : A_1 \times A_2 \rightarrow A_j$ is the *projection* function from the *product* set $A_1 \times A_2$ onto A_j . The *D-th power* of set A is the set A^D of all functions from set D to A . For each $d \in D$ there is the evaluation function $ev_d : A^D \rightarrow A$ having $ev_d(f) = f(d)$. The identity function on a set A is denoted $\text{id}_A : A \rightarrow A$.

The symbol $\circ \longrightarrow$ will be used for partial functions. Thus $f : A \circ \longrightarrow B$ means that f is a function with codomain B and domain $\text{Dom } f \subseteq A$. We may write $f(x) \downarrow$ to mean that $f(x)$ is defined, i.e. $x \in \text{Dom } f$. Associated with each insertion $\iota_j : A_j \rightarrow A_1 + A_2$ is its partial inverse, the *extraction* function $\varepsilon_j : A_1 + A_2 \circ \longrightarrow A_j$ having $\varepsilon_j(y) = x$ iff $\iota_j(x) = y$. Thus $y \in \text{Dom } \varepsilon_j$ iff $y \in \iota_j A_j$, i.e. $y = \iota_j(x)$ for some $x \in A_j$. Extraction functions play a vital role in the analysis of coalgebras built out of coproducts, as will be seen below.

Consider the following constructions of endofunctors $T : \mathbf{Set} \rightarrow \mathbf{Set}$.

- For a fixed set $D \neq \emptyset$, the *constant functor* \bar{D} has $\bar{D}(A) = D$ on sets A and $\bar{D}(f) = \text{id}_D$ on functions f .
- The *identity functor* Id has $\text{Id}A = A$ and $\text{Id}f = f$.
- The product $T_1 \times T_2$ of two functors has $T_1 \times T_2(A) = T_1A \times T_2A$, and, for a function $f : A \rightarrow B$, has $T_1 \times T_2(f)$ being the function

$$T_1(f) \times T_2(f) : T_1A \times T_2A \rightarrow T_1B \times T_2B$$

that acts by $(a_1, a_2) \mapsto (T_1(f)(a_1), T_2(f)(a_2))$.

- The coproduct $T_1 + T_2$ of two functors has $T_1 + T_2(A) = T_1A + T_2A$, and for $f : A \rightarrow B$, has $T_1 + T_2(f)$ being the function

$$T_1(f) + T_2(f) : T_1A + T_2A \rightarrow T_1B + T_2B$$

that acts by $\iota_j(a) \mapsto \iota_j(T_j(f)(a))$.

- The *D-th power functor* T^D of a functor T has $T^D A = (TA)^D$, and $T^D(f) : (TA)^D \rightarrow (TB)^D$ being the function $g \mapsto T(f) \circ g$.

A functor T is *polynomial* if it is constructed from constant functors and Id by finitely many applications of products, coproducts and powers. Note that any polynomial functor constructed without the use of Id is constant.

A *T-coalgebra* is a pair (A, α) comprising a set A and a function $A \xrightarrow{\alpha} TA$. A is the set of *states* and α is the *transition structure* of the coalgebra.

Note that A is determined as the domain $\text{Dom } \alpha$ of α , so we can identify the coalgebra with its transition structure, i.e. a T -coalgebra is any function of the form $\alpha : \text{Dom } \alpha \rightarrow T(\text{Dom } \alpha)$. A *morphism* from T -coalgebra α to T -coalgebra β is a function $f : \text{Dom } \alpha \rightarrow \text{Dom } \beta$ between their state sets which commutes with their transition structures in the sense that $\beta \circ f = T f \circ \alpha$, i.e. the following diagram commutes:

$$\begin{array}{ccc} \text{Dom } \alpha & \xrightarrow{f} & \text{Dom } \beta \\ \alpha \downarrow & & \downarrow \beta \\ T(\text{Dom } \alpha) & \xrightarrow{Tf} & T(\text{Dom } \beta) \end{array}$$

If $\text{Dom } \alpha \subseteq \text{Dom } \beta$, then α is a *subcoalgebra* of β iff the inclusion function $\text{Dom } \alpha \hookrightarrow \text{Dom } \beta$ is a morphism from α to β .

Every set $\{\alpha_i : i \in I\}$ of T -coalgebras has a *disjoint union* $\sum_I \alpha_i$, which is a T -coalgebra whose domain is the disjoint union of the $\text{Dom } \alpha_i$'s and whose transition structure acts as α_j on the summand $\iota_j \text{Dom } \alpha_j$ of $\text{Dom } \sum_I \alpha_i$. More precisely, this transition is given by $\iota_j(a) \mapsto T(\iota_j)(\alpha_j(a))$, with the insertion $\iota_j : \text{Dom } \alpha_j \rightarrow \text{Dom } \sum_I \alpha_i$ being an injective morphism making α_j isomorphic to a subcoalgebra of the disjoint union (see [17, Section 4]).

3 Syntax of Types, Terms and Formulas

Types

Fix a set \mathbb{O} of symbols called *observable types*, and a collection $\{\llbracket o \rrbracket : o \in \mathbb{O}\}$ of sets indexed by \mathbb{O} . Members of $\llbracket o \rrbracket$ are *observable elements*, or *constants*, of type o .

Example: $\mathbb{O} = \{\text{num}, \text{bool}, 1, 0\}$, with $\llbracket \text{num} \rrbracket = \{0, 1, \dots\}$,
 $\llbracket \text{bool} \rrbracket = \{\text{true}, \text{false}\}$, $\llbracket 1 \rrbracket = \{0\}$, $\llbracket 0 \rrbracket = \emptyset$.

The set of *types over* \mathbb{O} , or \mathbb{O} -*types*, is the smallest set \mathbb{T} such that $\mathbb{O} \subseteq \mathbb{T}$, $\text{St} \in \mathbb{T}$ and

- (1) if $\sigma_1, \sigma_2 \in \mathbb{T}$ then $\sigma_1 \times \sigma_2, \sigma_1 + \sigma_2 \in \mathbb{T}$;
- (2) if $\sigma \in \mathbb{T}$ and $o \in \mathbb{O}$, then $o \Rightarrow \sigma \in \mathbb{T}$.

A *subtype* of an \mathbb{O} -type τ is any type that occurs in the formation of τ .

St is a type symbol that will denote the state set of a given coalgebra. A type is *rigid* if it does not have St as a subtype. The set of rigid types is thus the smallest set that includes \mathbb{O} and satisfies (1) and (2).

The symbol “ o ” will always be reserved for members of \mathbb{O} . $o \Rightarrow \sigma$ is a power type: such types will always have an observable exponent.

Given any set A , we associate a set $\llbracket \sigma \rrbracket_A$ with each \mathbb{O} -type by putting $\llbracket o \rrbracket_A = \llbracket o \rrbracket$, $\llbracket \text{St} \rrbracket_A = A$, and inductively

$$\begin{aligned}\llbracket \sigma_1 \times \sigma_2 \rrbracket_A &= \llbracket \sigma_1 \rrbracket_A \times \llbracket \sigma_2 \rrbracket_A \\ \llbracket \sigma_1 + \sigma_2 \rrbracket_A &= \llbracket \sigma_1 \rrbracket_A + \llbracket \sigma_2 \rrbracket_A \\ \llbracket o \Rightarrow \sigma \rrbracket_A &= \llbracket \sigma \rrbracket_A^{\llbracket o \rrbracket}.\end{aligned}$$

If σ is a rigid type, then $\llbracket \sigma \rrbracket_A$ is a fixed set whose definition does not depend on A , so it may be written $\llbracket \sigma \rrbracket$.

Terms

To define *terms* we fix a denumerable set Var of *variables* and define a *context* to be a finite (possibly empty) list

$$\Gamma = (v_1 : \sigma_1, \dots, v_n : \sigma_n)$$

of assignments of \mathbb{O} -types σ_i to variables v_i , with the proviso that v_1, \dots, v_n are all distinct. Γ is a *rigid* context if all of the σ_i 's are rigid types. Concatenation of lists Γ and Γ' with disjoint sets of variables is written Γ, Γ' . A *term-in-context* is an expression of the form

$$\Gamma \triangleright M : \sigma,$$

which signifies that M is a “raw” term of type σ in context Γ . This may be abbreviated to $\Gamma \triangleright M$ if the type of the term is understood.

Figure 2 gives axioms that legislate terms into existence, and rules for generating new terms from given ones. The rules for products, coproducts and powers are the standard ones for introduction and elimination of terms of those types. Axiom (Con) states that an observable element is a constant term of its type, while the raw term \mathbf{s} in axiom (St) is a special parameter which will be interpreted as the “current” state in a coalgebra.

Bindings of variables in raw terms occur in lambda-abstractions and **case** terms: the v in the consequent of rule (Abs) and the v_j 's in the consequent of (Case) are bound in those terms. It is readily shown that in any term $\Gamma \triangleright \varphi$, all free variables of M appear in the list Γ . A *ground* term is one of the form $\emptyset \triangleright M : \sigma$, which may be abbreviated to the raw term M . Thus a ground term has no free variables. Note that a ground term may contain the state parameter \mathbf{s} , which behaves as a variable taking values in $\text{Dom } \alpha$.

A term is defined to be *rigid* if its context is rigid. This entails that every free variable of M is assigned a rigid type in Γ , and prevents any free variable of M from occurring in a subterm of type **St**. Of course all ground terms are rigid.

Axioms		
(Var) $\frac{v \in Var}{v : \sigma \triangleright v : \sigma}$	(Con) $\frac{c \in \llbracket o \rrbracket}{\emptyset \triangleright c : o}$	(St) $\frac{}{\emptyset \triangleright s : \mathbf{St}}$
Weakening		
(Weak) $\frac{\Gamma, \Gamma' \triangleright M : \sigma}{\Gamma, v : \sigma', \Gamma' \triangleright M : \sigma}$ where v does not occur in Γ or Γ' .		
Product Types		
(Pair) $\frac{\Gamma \triangleright M_1 : \sigma_1 \quad \Gamma \triangleright M_2 : \sigma_2}{\Gamma \triangleright \langle M_1, M_2 \rangle : \sigma_1 \times \sigma_2}$		
<div style="display: flex; justify-content: space-around;"> <div style="width: 45%;"> (Proj₁) $\frac{\Gamma \triangleright M : \sigma_1 \times \sigma_2}{\Gamma \triangleright \pi_1 M : \sigma_1}$ </div> <div style="width: 45%;"> (Proj₂) $\frac{\Gamma \triangleright M : \sigma_1 \times \sigma_2}{\Gamma \triangleright \pi_2 M : \sigma_2}$ </div> </div>		
Coproduct Types		
<div style="display: flex; justify-content: space-around;"> <div style="width: 45%;"> (In₁) $\frac{\Gamma \triangleright M : \sigma_1}{\Gamma \triangleright \iota_1 M : \sigma_1 + \sigma_2}$ </div> <div style="width: 45%;"> (In₂) $\frac{\Gamma \triangleright M : \sigma_2}{\Gamma \triangleright \iota_2 M : \sigma_1 + \sigma_2}$ </div> </div>		
(Case) $\frac{\Gamma \triangleright N : \sigma_1 + \sigma_2 \quad \Gamma, v_1 : \sigma_1 \triangleright M_1 : \sigma \quad \Gamma, v_2 : \sigma_2 \triangleright M_2 : \sigma}{\Gamma \triangleright \text{case } N \text{ of } [\iota_1 v_1 \mapsto M_1 \mid \iota_2 v_2 \mapsto M_2] : \sigma}$		
Power Types		
<div style="display: flex; justify-content: space-around;"> <div style="width: 45%;"> (Abs) $\frac{\Gamma, v : o \triangleright M : \sigma}{\Gamma \triangleright (\lambda v. M) : o \Rightarrow \sigma}$ </div> <div style="width: 45%;"> (App) $\frac{\Gamma \triangleright M : o \Rightarrow \sigma \quad \Gamma \triangleright N : o}{\Gamma \triangleright M \cdot N : \sigma}$ </div> </div>		

Fig. 2. Axioms and Rules for Generating Terms

τ -Terms

For a given \mathbb{O} -type τ , a τ -term is any term that can be generated by the axioms and rules of Figure 2 together with the additional rule

$$(Tr) \quad \frac{\Gamma \triangleright M : St}{\Gamma \triangleright \text{tr}(M) : \tau} .$$

Equations	
(Eq)	$\frac{\Gamma \triangleright M_1 : \sigma \quad \Gamma \triangleright M_2 : \sigma}{\Gamma \triangleright M_1 \approx M_2}$
Weakening	
(Weak)	$\frac{\Gamma, \Gamma' \triangleright \varphi}{\Gamma, v : \sigma', \Gamma' \triangleright \varphi} \quad \text{where } v \text{ does not occur in } \Gamma \text{ or } \Gamma'.$
Connectives	
(Neg)	$\frac{\Gamma \triangleright \varphi}{\Gamma \triangleright \neg \varphi}$
(Con)	$\frac{\Gamma \triangleright \varphi_1 \quad \Gamma \triangleright \varphi_2}{\Gamma \triangleright \varphi_1 \wedge \varphi_2}$

Fig. 3. Formation Rules for Formulas

Note that from this rule and the axiom (St) we can derive the τ -term

$$\emptyset \triangleright \text{tr}(\mathbf{s}) : \tau.$$

The symbol tr will denote the transition structure of coalgebras of the form $A \xrightarrow{\alpha} \llbracket \tau \rrbracket_A$. If M is interpreted as the state x of α , then $\text{tr}(M)$ is interpreted as $\alpha(x)$.

τ -Formulas

An *equation-in-context* has the form $\Gamma \triangleright M_1 \approx M_2$ where $\Gamma \triangleright M_1$ and $\Gamma \triangleright M_2$ are terms of the same type. A *formula-in-context* has the form $\Gamma \triangleright \varphi$, with the expression φ being constructed from equations $M_1 \approx M_2$ by propositional connectives. Formation rules for formulas are given in Figure 3, using the connectives \neg and \wedge . The other standard connectives \vee , \rightarrow , and \leftrightarrow can be introduced as definitional abbreviations in the usual way. A formula $\emptyset \triangleright \varphi$ with empty context is *ground*, and may be abbreviated to the expression φ . A *rigid* formula is one whose context is rigid.

A τ -*formula* is one that is generated by using only τ -terms as premisses in the rule (Eq). An *observable* formula is one that uses only terms of observable type in forming its component equations.

4 Semantics of Terms and Formulas

Each \mathbb{O} -type σ determines a polynomial functor $|\sigma| : \mathbf{Set} \rightarrow \mathbf{Set}$. For $o \in \mathbb{O}$, $|o|$ is the constant functor \bar{D} where $D = \llbracket o \rrbracket$; $|\mathbf{St}|$ is the identity functor Id ;

and inductively

$$|\sigma_1 \times \sigma_2| = |\sigma_1| \times |\sigma_2|, \quad |\sigma_1 + \sigma_2| = |\sigma_1| + |\sigma_2|, \quad |o \Rightarrow \sigma| = |\sigma|^{\llbracket o \rrbracket}.$$

Then in general, $|\sigma|A = \llbracket \sigma \rrbracket_A$ as defined earlier in Section 3. If σ is a rigid type, then $|\sigma|$ is the constant functor $|\sigma|A = \llbracket \sigma \rrbracket$.

A τ -coalgebra is a coalgebra for the functor $|\tau|$. A given τ -coalgebra $\alpha : A \rightarrow |\tau|A$ interprets types σ and contexts $\Gamma = (v_1 : \sigma_1, \dots, v_n : \sigma_n)$ by putting $\llbracket \sigma \rrbracket_\alpha = |\sigma|(\text{Dom } \alpha) = \llbracket \sigma \rrbracket_A$, and

$$\llbracket \Gamma \rrbracket_\alpha = \llbracket \sigma_1 \rrbracket_\alpha \times \dots \times \llbracket \sigma_n \rrbracket_\alpha$$

(so $\llbracket \emptyset \rrbracket_\alpha$ is the empty product 1). Hence α itself is a function of the form $A \rightarrow \llbracket \tau \rrbracket_\alpha$. The *denotation* of each τ -term $\Gamma \triangleright M : \sigma$, relative to the coalgebra α , is a function

$$\llbracket \Gamma \triangleright M : \sigma \rrbracket_\alpha : A \times \llbracket \Gamma \rrbracket_\alpha \longrightarrow \llbracket \sigma \rrbracket_\alpha,$$

defined by induction on the formation of terms. For empty contexts,

$$A \times \llbracket \emptyset \rrbracket_\alpha = A \times 1 \cong A,$$

so we replace $A \times \llbracket \emptyset \rrbracket_\alpha$ by A itself and interpret a ground term $\emptyset \triangleright M : \sigma$ as a function $A \rightarrow \llbracket \sigma \rrbracket_\alpha$.

Var:

$\llbracket v : \sigma \triangleright v : \sigma \rrbracket_\alpha : A \times \llbracket \sigma \rrbracket_\alpha \rightarrow \llbracket \sigma \rrbracket_\alpha$ is the right projection function.

Con:

$\llbracket \emptyset \triangleright c : o \rrbracket_\alpha : A \rightarrow \llbracket o \rrbracket$ is the constant function with value c .

St:

$\llbracket \emptyset \triangleright s : \text{St} \rrbracket_\alpha : A \rightarrow \llbracket \text{St} \rrbracket_\alpha$ is the identity function $A \rightarrow A$.

Tr:

$\llbracket \Gamma \triangleright \text{tr}(M) : \tau \rrbracket_\alpha : A \times \llbracket \Gamma \rrbracket_\alpha \rightarrow \llbracket \tau \rrbracket_\alpha$ is the composition of the functions

$$A \times \llbracket \Gamma \rrbracket_\alpha \xrightarrow{\llbracket \Gamma \triangleright M : \text{St} \rrbracket_\alpha} A \xrightarrow{\alpha} \llbracket \tau \rrbracket_\alpha.$$

Weak:

$\llbracket \Gamma, v : \sigma', \Gamma' \triangleright M : \sigma \rrbracket_\alpha$ is the composition of $\llbracket \Gamma, \Gamma' \triangleright M : \sigma \rrbracket_\alpha$ with the projection

$$A \times \llbracket \Gamma \rrbracket_\alpha \times \llbracket \sigma' \rrbracket_\alpha \times \llbracket \Gamma' \rrbracket_\alpha \longrightarrow A \times \llbracket \Gamma \rrbracket_\alpha \times \llbracket \Gamma' \rrbracket_\alpha.$$

Pair:

$\llbracket \Gamma \triangleright \langle M_1, M_2 \rangle : \sigma_1 \times \sigma_2 \rrbracket_\alpha$ is the product map

$$A \times \llbracket \Gamma \rrbracket_\alpha \xrightarrow{\langle \llbracket \Gamma \triangleright M_1 : \sigma_1 \rrbracket_\alpha, \llbracket \Gamma \triangleright M_2 : \sigma_2 \rrbracket_\alpha \rangle} \llbracket \sigma_1 \rrbracket_\alpha \times \llbracket \sigma_2 \rrbracket_\alpha.$$

Proj_j:

$\llbracket \Gamma \triangleright \pi_j M : \sigma_j \rrbracket_\alpha$ is the composition of

$$A \times \llbracket \Gamma \rrbracket_\alpha \xrightarrow{\llbracket \Gamma \triangleright M : \sigma_1 \times \sigma_2 \rrbracket_\alpha} \llbracket \sigma_1 \rrbracket_\alpha \times \llbracket \sigma_2 \rrbracket_\alpha \xrightarrow{\pi_j} \llbracket \sigma_j \rrbracket_\alpha.$$

Inj_j:

$\llbracket \Gamma \triangleright \iota_j M : \sigma_1 + \sigma_2 \rrbracket_\alpha$ is the composition of

$$A \times \llbracket \Gamma \rrbracket_\alpha \xrightarrow{\llbracket \Gamma \triangleright M : \sigma_j \rrbracket_\alpha} \llbracket \sigma_j \rrbracket_\alpha \xrightarrow{\iota_j} \llbracket \sigma_1 \rrbracket_\alpha + \llbracket \sigma_2 \rrbracket_\alpha.$$

Case:

This is most readily described at the level of function values. For $x \in A$ and $\gamma \in \llbracket \Gamma \rrbracket_\alpha$, let

$$\llbracket \Gamma \triangleright N : \sigma_1 + \sigma_2 \rrbracket_\alpha(x, \gamma) = \iota_j(a) \in \llbracket \sigma_1 \rrbracket_\alpha + \llbracket \sigma_2 \rrbracket_\alpha$$

(which holds for a unique j and $a \in \llbracket \sigma_j \rrbracket_\alpha$). Then the element

$$\llbracket \Gamma \triangleright \text{case } N \text{ of } [\iota_1 v_1 \mapsto M_1 \mid \iota_2 v_2 \mapsto M_2] : \sigma \rrbracket_\alpha(x, \gamma)$$

of $\llbracket \sigma \rrbracket_\alpha$ is defined to be

$$\llbracket \Gamma, v_j : \sigma_j \triangleright M_j : \sigma \rrbracket_\alpha(x, \gamma, a).$$

Abs:

$\llbracket \Gamma \triangleright (\lambda v. M) : o \Rightarrow \sigma \rrbracket_\alpha(x, \gamma)$ is the function $\llbracket o \rrbracket \rightarrow \llbracket \sigma \rrbracket_\alpha$ given by

$$a \mapsto \llbracket \Gamma, v : o \triangleright M : \sigma \rrbracket_\alpha(x, \gamma, a).$$

App:

$\llbracket \Gamma \triangleright M \cdot N : \sigma \rrbracket_\alpha(x, \gamma)$ is the element of $\llbracket \sigma \rrbracket_\alpha$ obtained by evaluating the function

$$\llbracket \Gamma \triangleright M : o \Rightarrow \sigma \rrbracket_\alpha(x, \gamma) : \llbracket o \rrbracket \longrightarrow \llbracket \sigma \rrbracket_\alpha$$

at $\llbracket \Gamma \triangleright N : o \rrbracket_\alpha(x, \gamma) \in \llbracket o \rrbracket$.

This completes the inductive definition of $\llbracket \Gamma \triangleright M : \sigma \rrbracket_\alpha$.

Semantics of Formulas

A τ -equation $\Gamma \triangleright M_1 \approx M_2$ is said to be *valid* in coalgebra α if the α -denotations $\llbracket \Gamma \triangleright M_1 \rrbracket_\alpha$ and $\llbracket \Gamma \triangleright M_2 \rrbracket_\alpha$ of the terms $\Gamma \triangleright M_j$ are identical. More generally we introduce a satisfaction relation

$$\alpha, x, \gamma \models \Gamma \triangleright \varphi,$$

for τ -formulas in τ -coalgebras, which expresses that $\Gamma \triangleright \varphi$ is *satisfied*, or *true*, in α at state x under the value-assignment γ to the variables of context Γ . This

is defined inductively by

$$\begin{aligned} \alpha, x, \gamma \models \Gamma \triangleright M_1 \approx M_2 & \text{ iff } \llbracket \Gamma \triangleright M_1 \rrbracket_\alpha(x, \gamma) = \llbracket \Gamma \triangleright M_2 \rrbracket_\alpha(x, \gamma), \\ \alpha, x, \gamma \models \Gamma \triangleright \neg \varphi & \text{ iff not } \alpha, x, \gamma \models \Gamma \triangleright \varphi, \\ \alpha, x, \gamma \models \Gamma \triangleright \varphi_1 \wedge \varphi_2 & \text{ iff } \alpha, x, \gamma \models \Gamma \triangleright \varphi_1 \text{ and } \alpha, x, \gamma \models \Gamma \triangleright \varphi_2. \end{aligned}$$

$\Gamma \triangleright \varphi$ is *true at x* , written $\alpha, x \models \Gamma \triangleright \varphi$, if $\alpha, x, \gamma \models \Gamma \triangleright \varphi$ for all $\gamma \in \Gamma$. α is a *model of $\Gamma \triangleright \varphi$* , written $\alpha \models \Gamma \triangleright \varphi$, if $\alpha, x \models \Gamma \triangleright \varphi$ for all states $x \in \text{Dom } \alpha$. In that case we also say that $\Gamma \triangleright \varphi$ is *valid in the coalgebra α* .

Substitution

In working with this system it becomes essential to have available the operation $N[M/v]$ of substituting the raw term M for the variable v in N . The following rule is derivable:

$$\frac{\Gamma \triangleright M : \sigma \quad \Gamma, v : \sigma \triangleright N : \sigma'}{\Gamma \triangleright N[M/v] : \sigma'}$$

The semantics of terms obeys the basic principle that substitution is interpreted as *composition* of denotations [13, 2.2]. Because of the special role of the state set A , this takes the form

$$\llbracket \Gamma \triangleright N[M/v] \rrbracket_\alpha = \llbracket \Gamma, v : \sigma \triangleright N \rrbracket_\alpha \circ \langle \pi_1, \pi_2, \llbracket \Gamma \triangleright M \rrbracket_\alpha \rangle,$$

so that the diagram

$$\begin{array}{ccc} A \times \llbracket \Gamma \rrbracket_\alpha & \xrightarrow{\langle \pi_1, \pi_2, \llbracket M \rrbracket_\alpha \rangle} & A \times \llbracket \Gamma \rrbracket_\alpha \times \llbracket \sigma \rrbracket_\alpha \\ & \searrow \llbracket N[M/v] \rrbracket_\alpha & \downarrow \llbracket N \rrbracket_\alpha \\ & & \llbracket \sigma' \rrbracket_\alpha \end{array}$$

commutes. It is also possible to make substitutions $N[M/s]$ for the state parameter s according to the rule

$$\frac{\Gamma \triangleright M : \text{St} \quad \Gamma \triangleright N : \sigma'}{\Gamma \triangleright N[M/s] : \sigma'}$$

with the semantics $\llbracket \Gamma \triangleright N[M/s] \rrbracket_\alpha = \llbracket \Gamma \triangleright N \rrbracket_\alpha \circ \langle \llbracket \Gamma \triangleright M \rrbracket_\alpha, \pi_2 \rangle :$

$$\begin{array}{ccc} A \times \llbracket \Gamma \rrbracket_\alpha & \xrightarrow{\langle \llbracket \Gamma \triangleright M \rrbracket_\alpha, \pi_2 \rangle} & A \times \llbracket \Gamma \rrbracket_\alpha \\ & \searrow \llbracket N[M/s] \rrbracket_\alpha & \downarrow \llbracket N \rrbracket_\alpha \\ & & \llbracket \sigma' \rrbracket_\alpha \end{array}$$

5 Paths and Bisimulations

If (A, α) and (B, β) are coalgebras for a functor T , then a relation $R \subseteq A \times B$ is a T -bisimulation from α to β if there exists a transition structure $\rho : R \rightarrow TR$ on R such that the projections from R to A and B are coalgebraic morphisms from ρ to α and β , i.e. the following diagram commutes:

$$\begin{array}{ccccc} A & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & B \\ \alpha \downarrow & & \downarrow \rho & & \downarrow \beta \\ TA & \xleftarrow{T\pi_1} & TR & \xrightarrow{T\pi_2} & TB \end{array}$$

A function $f : A \rightarrow B$ is a morphism from α to β iff its graph $\{(a, f(a)) : a \in A\}$ is a bisimulation from α to β [17, Theorem 2.5]: a morphism is essentially a functional bisimulation. When $\text{Dom } \alpha \subseteq \text{Dom } \beta$, α is a subcoalgebra of β iff the identity relation on $\text{Dom } \alpha$ is a bisimulation from α to β .

The above categorical definition of bisimulation appeared in [1]. It has a characterisation in terms of “liftings” of relations [5,6]. For $R \subseteq A \times B$, define a relation $R^T \subseteq TA \times TB$ by induction on the formation of the polynomial functor T :

$$\begin{aligned} R^{\bar{D}} &= \text{id}_D \\ R^{\text{Id}} &= R \\ R^{T_1 \times T_2} &= \{(x, y) : \pi_1 x R^{T_1} \pi_1 y \text{ and } \pi_2 x R^{T_2} \pi_2 y\} \\ R^{T_1 + T_2} &= \{(\iota_1 x, \iota_1 y) : x R^{T_1} y\} \cup \{(\iota_2 x, \iota_2 y) : x R^{T_2} y\} \\ R^{T^D} &= \{(f, g) : \forall d \in D \ f(d) R^T g(d)\}. \end{aligned}$$

These liftings preserve many basic properties of relations. Thus if R is total ($\text{Dom } R = A$) or surjective (onto B) or injective or functional, then R^T will also have the corresponding property.

Theorem 5.1 (Folklore)

If $R \subseteq \text{Dom } \alpha \times \text{Dom } \beta$, where α and β are T -coalgebras, then R is a bisimulation from α to β if, and only if,

$$x R y \text{ implies } \alpha(x) R^T \beta(y)$$

for all states x in α and y in β . □

The inverse of a bisimulation is a bisimulation, and the union of any collection of bisimulations from α to β is a bisimulation [17, Section 5]. Hence there is a largest bisimulation from α to β , which is a *symmetric* relation called *bisimilarity*. We denote this by \sim . States x and y are *bisimilar*, $x \sim y$, when

xRy for some bisimulation R between α and β . This is intended to capture the notion that x and y are observationally indistinguishable.

Theorem 5.1 can be used to show that bisimulations preserve the values of terms, and in particular that related states assign the same values to observable terms. To explain this we need some more notation. Let (A, α) and (B, β) be τ -coalgebras, and $R \subseteq A \times B$. Then for each \mathbb{O} -type σ we have the lifted relation $R^{|\sigma|} \subseteq \llbracket \sigma \rrbracket_A \times \llbracket \sigma \rrbracket_B$, where $|\sigma|$ is the functor defined by σ . For observable σ , $R^{|\sigma|}$ is just the identity relation on $\llbracket \sigma \rrbracket$. The same is true whenever σ is a rigid type. For any context $\Gamma = (v_1 : \sigma_1, \dots, v_n : \sigma_n)$ we define a relation $R^\Gamma \subseteq \llbracket \Gamma \rrbracket_A \times \llbracket \Gamma \rrbracket_B$ as the direct product of the $R^{|\sigma_i|}$'s, i.e.

$$(\gamma_1, \dots, \gamma_n) R^\Gamma (\gamma'_1, \dots, \gamma'_n) \text{ iff } \gamma_i R^{|\sigma_i|} \gamma'_i \text{ for all } i \leq n.$$

For rigid Γ , R^Γ is just the identity relation on $\llbracket \Gamma \rrbracket = \llbracket \sigma_1 \rrbracket \times \dots \times \llbracket \sigma_n \rrbracket$.

Theorem 5.2 (Value-Preservation)

Let R be a $|\tau|$ -bisimulation from α to β .

(1) For any τ -term $\Gamma \triangleright M : \sigma$, if $\gamma \in \llbracket \Gamma \rrbracket_\alpha$ and $\gamma' \in \llbracket \Gamma \rrbracket_\beta$ have $\gamma R^\Gamma \gamma'$, then

$$xRy \text{ implies } \llbracket \Gamma \triangleright M \rrbracket_\alpha(x, \gamma) R^{|\sigma|} \llbracket \Gamma \triangleright M \rrbracket_\beta(y, \gamma').$$

(2) If $\Gamma \triangleright M$ is a term of observable type, and $\gamma R^\Gamma \gamma'$, then

$$xRy \text{ implies } \llbracket \Gamma \triangleright M \rrbracket_\alpha(x, \gamma) = \llbracket \Gamma \triangleright M \rrbracket_\beta(y, \gamma').$$

(3) If $\Gamma \triangleright M$ is a rigid term of observable type, and $\gamma \in \llbracket \Gamma \rrbracket$, then

$$xRy \text{ implies } \llbracket \Gamma \triangleright M \rrbracket_\alpha(x, \gamma) = \llbracket \Gamma \triangleright M \rrbracket_\beta(y, \gamma). \quad \square$$

From part (2) of this result it follows, by induction on the formation of formulas, that if $\Gamma \triangleright \varphi$ is an *observable* formula, then

$$\alpha, x, \gamma \models \Gamma \triangleright \varphi \text{ iff } \beta, y, \gamma' \models \Gamma \triangleright \varphi$$

whenever xRy and $\gamma R^\Gamma \gamma'$. Thus if $\Gamma \triangleright \varphi$ is valid in α and R is surjective, so that R^Γ is also surjective, $\Gamma \triangleright \varphi$ will be valid in β . On the other hand if $\beta \models \Gamma \triangleright \varphi$ and R is total, so that R^Γ is also total, then $\alpha \models \Gamma \triangleright \varphi$. In other words, validity is preserved in passing from α to β if β is the image of a bisimulation from α , and is preserved in passing from β to α if α is the domain of a bisimulation to β . If $\Gamma \triangleright \varphi$ is also rigid, then its validity is preserved by disjoint unions: given any element $\iota_j(a)$ of $\sum_I \alpha_i$ and any $\gamma \in \llbracket \Gamma \rrbracket$, if $\alpha_j \models \Gamma \triangleright \varphi$ we get $\sum_I \alpha_i, \iota_j(a), \gamma \models \Gamma \triangleright \varphi$, because $\alpha_j, a, \gamma \models \Gamma \triangleright \varphi$, $\gamma R^\Gamma \gamma$, and the insertion morphism ι_j is a bisimulation. To sum up:

Theorem 5.3 The class $\{\alpha : \alpha \models \Gamma \triangleright \varphi\}$ of all models of an observable formula is closed under domains and images of bisimulations, including domains and images of morphisms as well as subcoalgebras. If $\Gamma \triangleright \varphi$ is rigid and observable, then its class of models is also closed under disjoint unions. \square

The main purpose of this Section is to strengthen Theorem 5.2 to a logical characterisation of bisimilarity: states are bisimilar when they assign the same

values to all ground terms of observable type, or equivalently when they satisfy the same rigid observable formulas (see Theorem 5.8). The key to this is the relation $\equiv_{\alpha\beta}$ defined by

$$x \equiv_{\alpha\beta} y \text{ iff } \llbracket M \rrbracket_{\alpha}(x) = \llbracket M \rrbracket_{\beta}(y) \text{ for all ground terms } M : o \\ \text{of observable type.}$$

$\equiv_{\alpha\beta}$ is a bisimulation from α to β , and turns out to be the largest one. The proof of this requires the development of another characterisation of bisimulation, using the notion of “paths” between functors [10, Section 6].

A *path* is a finite list of symbols of the kinds π_j , ε_j , ev_d . Write $p.q$ for concatenation of lists p and q . The notation $T \xrightarrow{p} S$ means that p is a path from functor T to functor S , and is defined by the following conditions

- $T \xrightarrow{\langle \rangle} T$, where $\langle \rangle$ is the empty path.
- $T_1 \times T_2 \xrightarrow{\pi_j \cdot p} S$ whenever $T_j \xrightarrow{p} S$, for $j = 1, 2$.
- $T_1 + T_2 \xrightarrow{\varepsilon_j \cdot p} S$ whenever $T_j \xrightarrow{p} S$, for $j = 1, 2$.
- $T^D \xrightarrow{ev_d \cdot p} S$ for all $d \in D$ whenever $T \xrightarrow{p} S$.

It is evident that for any path $T \xrightarrow{p} S$, S is one of the functors involved in the formation of T .

A path $T \xrightarrow{p} S$ induces a partial function $p_A : TA \multimap SA$ for each set A , defined by induction on the length of p as follows.

- $\langle \rangle_A : TA \multimap TA$ is the identity function id_{TA} , so is totally defined.
- $(\pi_j \cdot p)_A = p_A \circ \pi_j$, the composition of $T_1 A \times T_2 A \xrightarrow{\pi_j} TA \xrightarrow{p_A} SA$.
Thus $x \in \text{Dom}(\pi_j \cdot p)_A$ iff $\pi_j(x) \in \text{Dom } p_A$.
- $(\varepsilon_j \cdot p)_A = p_A \circ \varepsilon_j$, the composition of $T_1 A + T_2 A \xrightarrow{\varepsilon_j} TA \xrightarrow{p_A} SA$.
Thus $x \in \text{Dom}(\varepsilon_j \cdot p)_A$ iff $x \in \iota_j TA_j$ and $\varepsilon_j(x) \in \text{Dom } p_A$.
- $(ev_d \cdot p)_A = p_A \circ ev_d$, the composition of $(TA)^D \xrightarrow{ev_d} TA \xrightarrow{p_A} SA$.
Thus $f \in \text{Dom}(ev_d \cdot p)_A$ iff $f(d) \in \text{Dom } p_A$.

A path $T \xrightarrow{p} S$ is a *state path* if $S = \text{Id}$, and an *observation path* if $S = \bar{D}$ for some set D . A T -bisimulation can be characterised as a relation that is “preserved” by the partial functions induced by state and observation paths from T . To explain this we adopt the convention that whenever we write “ $f(x) Q g(y)$ ” for some relation Q and some partial functions f and g we mean that

$f(x)$ is defined iff $g(y)$ is defined, and $(f(x), g(y)) \in Q$ when they are both defined.

Theorem 5.4 *Let $R \subseteq A \times B$, $x \in TA$, and $y \in TB$, where T is a polynomial*

functor. Then the following are equivalent.

- (1) xR^Ty .
- (2) For all paths $T \xrightarrow{p} S$, $p_A(x)R^Sp_B(y)$.
- (3) • For all state paths $T \xrightarrow{p} \text{Id}$, $p_A(x)Rp_B(y)$; and
 • for all observation paths $T \xrightarrow{p} \bar{D}$, $p_A(x) = p_B(y)$. □

Combining this result with Theorem 5.1 gives the desired “dynamic” characterisation of bisimulations:

Theorem 5.5 *If $A \xrightarrow{\alpha} TA$ and $B \xrightarrow{\beta} TB$ are coalgebras for a polynomial functor T , then a relation $R \subseteq A \times B$ is a T -bisimulation if, and only if, xRy implies*

- for all state paths $T \xrightarrow{p} \text{Id}$, $p_A(\alpha(x))Rp_B(\beta(y))$; and
- for all observation paths $T \xrightarrow{p} \bar{D}$, $p_A(\alpha(x)) = p_B(\beta(y))$.

Corollary 5.6 *If $C \subseteq \text{Dom } \alpha$, then C is a subcoalgebra of α iff $x \in C$ implies $p_A(\alpha(x)) \in C$ for all state paths $T \xrightarrow{p} \text{Id}$ such that $p_A(\alpha(x)) \downarrow$.*

Proof. To say that C is a subcoalgebra of α means that there is some T -transition structure on C that is a subcoalgebra of α . Such a structure is unique, and exists iff the identity relation $\Delta_C = \{(x, x) : x \in C\}$ on C is a bisimulation relation on α [17, Proposition 6.2]. Now apply the Theorem with $R = \Delta_C$ and $\alpha = \beta$, and use the fact that $p_A(\alpha(x))\Delta_C p_A(\alpha(x))$ iff $p_A(\alpha(x)) \in C$. □

This characterisation makes it easy to see that if R is a bisimulation from α to β , then $\text{Dom } R$ is a subcoalgebra of α . For if $x \in \text{Dom } R$ and $p_A(\alpha(x)) \downarrow$, then xRy for some y , so $p_A(\alpha(x))Rp_B(\beta(y))$ by 5.5 and hence $p_A(\alpha(x)) \in \text{Dom } R$. Similarly, the *image* of R is seen to be a subcoalgebra of β .

Path functions are thus an effective tool in the structural analysis of polynomial coalgebras. Their use in logical characterisations derives from the fact that the action of a path function is definable by a (ground) term.

Lemma 5.7 (Path Lemma)

For any path $|\tau| \xrightarrow{p} |\sigma|$ there exists a term of the form

$$v : \tau \triangleright \bar{p} : \sigma$$

such that for any τ -coalgebra (A, α) and any $x \in A$, if $\alpha(x) \in \text{Dom } p_A$ then

$$p_A(\alpha(x)) = \llbracket \bar{p}[\text{tr}(\mathbf{s})/v] \rrbracket_\alpha(x). \quad \square$$

Note that since $\emptyset \triangleright \text{tr}(\mathbf{s}) : \tau$ is a τ -theorem, so too is $\emptyset \triangleright \bar{p}[\text{tr}(\mathbf{s})/v] : \sigma$ by the rule of Substitution. Hence $\bar{p}[\text{tr}(\mathbf{s})/v]$ is a ground term of type σ .

In proving the Path Lemma (by induction on the length of p) it must be shown that the action of an extraction function ε_j is term-definable. In fact it can be shown that for any term $\Gamma \triangleright M : \sigma_1 + \sigma_2$ of coproduct type there exist terms $\Gamma \triangleright \varepsilon_j M : \sigma_j$ for $j = 1, 2$ such that

$$\llbracket \Gamma \triangleright \varepsilon_j M \rrbracket_\alpha(x, \gamma) = \varepsilon_j(\llbracket \Gamma \triangleright M \rrbracket_\alpha(x, \gamma)) \in \llbracket \sigma_j \rrbracket_A$$

whenever $\llbracket \Gamma \triangleright M \rrbracket_\alpha(x, \gamma) \in \iota_j \llbracket \sigma_j \rrbracket_A$. Indeed, taking v_1, v_2 as new variables not in M , put

$$\varepsilon_1 M := \text{case } M \text{ of } [\iota_1 v_1 \mapsto v_1 \mid \iota_2 v_2 \mapsto N_1],$$

where N_1 is any ground term of type σ_1 (the existence of ground terms of every type follows by induction on term and type formation from axioms (Con) and (St) of Figure 2). ε_2 is defined similarly.

The term function $\llbracket \bar{p}[\text{tr}(\mathbf{s})/v] \rrbracket_\alpha$ has domain A , and so may not be identical to $p_A \circ \alpha$ if p_A is partial. This is only an issue when the path p includes an extraction symbol ε_j (for otherwise p_A is total), but further use of **case** allows the construction of terms that “discriminate” between the two summands of a coproduct $\llbracket \sigma_1 \rrbracket_A + \llbracket \sigma_2 \rrbracket_A$ and determine whether $p_A(\alpha(x))$ is defined. For this to work we need the (reasonable) assumption that τ has at least one observable subtype o that is *non-trivial* in the sense that $\llbracket o \rrbracket$ has at least two distinct members, say c_1 and c_2 . Then we form the term $v : \sigma_1 + \sigma_2 \triangleright P : o$, where

$$P := \text{case } v \text{ of } [\iota_1 v_1 \mapsto c_1 \mid \iota_2 v_2 \mapsto c_2],$$

and find, when α is a $\sigma_1 + \sigma_2$ -coalgebra, that the ground term $P[\text{tr}(\mathbf{s})/v] : o$ is a discriminator:

$$\llbracket P[\text{tr}(\mathbf{s})/v] \rrbracket_\alpha(x) = c_j \text{ iff } \alpha(x) \in \iota_j \llbracket \sigma_j \rrbracket_A = \text{Dom } \varepsilon_j.$$

An inductive argument that repeats this construction for each extraction symbol in a path $|\tau| \xrightarrow{p} |\sigma|$ produces a finite set T_p of ground observable terms such that

$$\text{if } (A, \alpha) \text{ and } (B, \beta) \text{ are } \tau\text{-coalgebras, and } x \in A \text{ and } y \in B \text{ have } \llbracket M \rrbracket_\alpha(x) = \llbracket M \rrbracket_\beta(y) \text{ for all } M \in T_p, \text{ then } \alpha(x) \in \text{Dom } p_A \text{ iff } \beta(y) \in \text{Dom } p_B.$$

Combining this observation with the Path Lemma 5.7, the path-characterisation of bisimulations of Theorem 5.5, and application of Substitution rules, leads ultimately to a proof that the relation $\equiv_{\alpha\beta}$ is a bisimulation. This in turn leads to the logical characterisation of bisimilarity of states:

Theorem 5.8 *Let (A, α) and (B, β) be τ -coalgebras, where τ has at least one non-trivial observable subtype. Then for any $x \in A$ and $y \in B$, the following are equivalent:*

- (1) x and y are bisimilar: $x \sim y$.

- (2) $\alpha, x \models \Gamma \triangleright \varphi$ iff $\beta, y \models \Gamma \triangleright \varphi$ for all rigid observable formulas $\Gamma \triangleright \varphi$.
- (3) $\alpha, x \models M \approx N$ implies $\beta, y \models M \approx N$ for all ground observable terms M and N .
- (4) $\llbracket M \rrbracket_\alpha(x) = \llbracket M \rrbracket_\beta(y)$ for all ground observable terms M , i.e. $x \equiv_{\alpha\beta} y$.

6 Observable Ultrapowers

Let U be an ultrafilter on a set I . For each set A , the relation

$$f =_U g \text{ iff } \{i \in I : f(i) = g(i)\} \in U$$

is an equivalence relation on the I -th power A^I of A . Each $f \in A^I$ has the equivalence class $f^U = \{g \in A^I : f =_U g\}$. The quotient set

$$A^U = \{f^U : f \in A^I\}$$

is called the *ultrapower of A with respect to U* .² There is a natural injection $e_A : A \rightarrow A^U$ given by $e_A(a) = \bar{a}^U$, where $\bar{a} \in A^I$ is the constant function with value a . The distinction between a and \bar{a}^U is sometimes blurred, allowing A to be identified with the subset $e_A(A)$ of A^U .

A notation that will be useful below is to write $f^U \in_U C$, for $C \subseteq A$, when $\{i \in I : f(i) \in C\} \in U$.

A map $\theta : A_1 \times \cdots \times A_n \rightarrow B$ has a *U -lifting* $\theta^U : A_1^U \times \cdots \times A_n^U \rightarrow B^U$, given by

$$\theta^U(f_1^U, \dots, f_n^U) = \langle \theta(f_1(i), \dots, f_n(i)) : i \in I \rangle^U.$$

In the case $n = 1$, any $\theta : A \rightarrow B$ lifts to $\theta^U : A^U \rightarrow B^U$ where $\theta^U(f^U) = (\theta \circ f)^U$. This works also for a partial $\theta : A \circ \longrightarrow B$, providing a U -lifting $\theta^U : A^U \circ \longrightarrow B^U$ in the same way, with the proviso that $\theta^U(f^U)$ is defined only when $f^U \in \text{Dom } \theta$, i.e. when $\{i \in I : f(i) \in \text{Dom } \theta\} \in U$.

Now let $\alpha : A \rightarrow \llbracket \tau \rrbracket_A$ be a τ -coalgebra which will remain fixed throughout Section 6. The transition structure α lifts to a function $\alpha^U : A^U \rightarrow \llbracket \tau \rrbracket_A^U$, and each term denotation $\llbracket \Gamma \triangleright M : \sigma \rrbracket_\alpha$ lifts to a function

$$\llbracket \Gamma \triangleright M : \sigma \rrbracket_\alpha^U : A^U \times \llbracket \sigma_1 \rrbracket_A^U \times \cdots \times \llbracket \sigma_n \rrbracket_A^U \longrightarrow \llbracket \sigma \rrbracket_A^U \quad (\ddagger)$$

where $\sigma_1, \dots, \sigma_n$ is the list of types of Γ .

α^U is not a τ -coalgebra on A^U since its codomain is $\llbracket \tau \rrbracket_A^U = (|\tau|(A))^U$ rather than $\llbracket \tau \rrbracket_{A^U} = |\tau|(A^U)$. We wish to define a coalgebraic structure on A^U that interprets terms in a manner related to the functions $\llbracket \Gamma \triangleright M : \sigma \rrbracket_\alpha^U$. To achieve this it is necessary to retain only some of the points of A^U , and the key to understanding which ones is provided by considering the U -lifting of the α -denotation of a ground observable term $M : o$. This is the function

² For the standard theory of ultrafilters and ultrapowers see [2,3].

$\llbracket M \rrbracket_\alpha^U : A^U \rightarrow \llbracket o \rrbracket^U$. To act as a denotation for M it should assign values in $\llbracket o \rrbracket$, viewed as a subset of $\llbracket o \rrbracket^U$. In other words we should have

$$\llbracket M \rrbracket_\alpha^U(x) \in e\llbracket o \rrbracket = \{\bar{c}^U : c \in \llbracket o \rrbracket\} \subseteq \llbracket o \rrbracket^U.$$

We are thus led to define an element x of A^U to be *observable* if $\llbracket M \rrbracket_\alpha^U(x) \in e\llbracket o \rrbracket$ for every ground observable τ -term $M : o$. If $x = f^U$, this means that for each such M there exists an observable element $c_M \in \llbracket o \rrbracket$ such that

$$\{i \in I : \llbracket M \rrbracket_\alpha(f(i)) = c_M\} \in U. \quad (\dagger)$$

Put $A^+ = \{x \in A^U : x \text{ is observable}\}$. For each $a \in A$ and any ground $M : o$,

$$\llbracket M \rrbracket_\alpha^U(e_A(a)) = \llbracket M \rrbracket_\alpha^U(\bar{a}^U) = (\llbracket M \rrbracket_\alpha \circ \bar{a})^U = \left(\overline{\llbracket M \rrbracket_\alpha(a)}\right)^U \in e\llbracket o \rrbracket,$$

so $e_A(a)$ is observable. Thus e_A embeds A into A^+ , allowing us to view A^+ as an extension of A .

Theorem 6.1 *For any path $|\tau| \xrightarrow{p} |\sigma|$ beginning at $|\tau|$ there exist partial functions $(p_A \circ \alpha)^+ : A^+ \multimap \llbracket \sigma \rrbracket_{A^+}$ and $\theta_\sigma : \llbracket \sigma \rrbracket_A^U \multimap \llbracket \sigma \rrbracket_{A^+}$,*

$$\begin{array}{ccccc} A & \xrightarrow{e_A} & A^+ & \hookrightarrow & A^U \\ \downarrow p_A \circ \alpha & & \downarrow (p_A \circ \alpha)^+ & & \downarrow (p_A \circ \alpha)^U \\ \llbracket \sigma \rrbracket_A & \xrightarrow{|\sigma|e_A} & \llbracket \sigma \rrbracket_{A^+} & \xleftarrow{\theta_\sigma} & \llbracket \sigma \rrbracket_A^U \end{array}$$

such that

- $\text{Dom}(p_A \circ \alpha)^+ = A^+ \cap \text{Dom}(p_A \circ \alpha)^U$;
- $x \in \text{Dom}(p_A \circ \alpha)^+ \text{ implies } (p_A \circ \alpha)^U(x) \in \text{Dom } \theta_\sigma$;
- $a \in \text{Dom } p_A \circ \alpha \text{ implies } e_A(a) \in \text{Dom}(p_A \circ \alpha)^+$;
- $b \in \llbracket \sigma \rrbracket_A \text{ implies } \bar{b}^U \in \text{Dom } \theta_\sigma$;
- $\theta_\sigma \text{ is surjective (onto } \llbracket \sigma \rrbracket_{A^+})$,

and the above diagram commutes wherever defined. \square

The proof of this theorem proceeds by induction on the formation of the end-type σ , and is too long and complex to be described here. But some comments are in order, particularly since the function θ_σ seems to be pointing in the “wrong” direction. When σ is observable, θ_σ is just the inverse of the embedding $\llbracket \sigma \rrbracket \hookrightarrow \llbracket \sigma \rrbracket^U$, and when $\sigma = \text{St}$, θ_σ is the inverse of the inclusion $A^+ \hookrightarrow A^U$. The inductive cases for products and coproducts appeal to the fact that the ultrapower operation commutes with these constructions, in the

sense that there exist isomorphisms

$$(B \times D)^U \cong B^U \times D^U, \quad (B + D)^U \cong B^U + D^U$$

for any sets B, D . However there is no corresponding commutation for powers: there is only a surjection $\chi : (B^D)^U \twoheadrightarrow (B^U)^D$, given by the formula $\chi(x)(d) = ev_d^U(x)$, which may not be injective (this uses the U -lifting of $ev_d : B^D \rightarrow D$). χ is used in the definition of θ_σ when σ is a power type, and this dictates the direction of θ_σ .

Now applying Theorem 6.1 in the case that $\sigma = \tau$ and p is the empty path, so that $p_A = \text{id}_A$, gives a function $\alpha^+ : A^+ \circ \rightarrow \llbracket \tau \rrbracket_{A^+}$ whose domain is $A^+ \cap \text{Dom } \alpha^U = A^+$, so that α^+ is total, such that the diagram

$$\begin{array}{ccccc} A & \xrightarrow{e_A} & A^+ & \hookrightarrow & A^U \\ \alpha \downarrow & & \downarrow \alpha^+ & & \downarrow \alpha^U \\ \llbracket \tau \rrbracket_A & \xrightarrow{|\tau|e_A} & \llbracket \tau \rrbracket_{A^+} & \xleftarrow{\theta_\tau \circ} & \llbracket \tau \rrbracket_A^U \end{array}$$

commutes. α^+ is thus a τ -coalgebra, which will be called the *observable ultra-power* of α with respect to U .

The use we make of α^+ derives ultimately from that fact that for a ground observable term $M : o$, the denotation $\llbracket M \rrbracket_{\alpha^+}$ agrees with $\llbracket M \rrbracket_\alpha^U$ in the sense that $\llbracket M \rrbracket_{\alpha^+} = \theta_o \circ \llbracket M \rrbracket_\alpha^U \upharpoonright_{A^+}$, or equivalently $\llbracket M \rrbracket_\alpha^U \upharpoonright_{A^+} = e \circ \llbracket M \rrbracket_{\alpha^+}$:

$$\begin{array}{ccc} & & \llbracket o \rrbracket^U \\ & \nearrow \llbracket M \rrbracket_\alpha^U & \uparrow e \\ A^+ & \xrightarrow{\llbracket M \rrbracket_{\alpha^+}} & \llbracket o \rrbracket \\ & & \downarrow \theta_o \end{array}$$

But to prove that takes an induction on the derivation of the ground term $\emptyset \triangleright M$, which may involve more complex types and non-empty contexts. Therefore we have to prove a more elaborate result. To formulate this, given a context Γ with types $\sigma_1, \dots, \sigma_n$, let $\theta_\Gamma = \theta_{\sigma_1} \times \dots \times \theta_{\sigma_n}$ be the product of the functions $\theta_{\sigma_i} : \llbracket \sigma_i \rrbracket_A^U \circ \rightarrow \llbracket \sigma_i \rrbracket_{A^+}$. Then $\text{Dom } \theta_\Gamma$ is the product of the $\text{Dom } \theta_{\sigma_i}$'s, and so $A^+ \times \text{Dom } \theta_\Gamma$ is a subset of the domain of $\llbracket \Gamma \triangleright M \rrbracket_\alpha^U$ for any term in context Γ (see (§) earlier in this section for $\text{Dom } \llbracket \Gamma \triangleright M \rrbracket_\alpha^U$).

We can now state the result that explains the sense in which $\llbracket \Gamma \triangleright M \rrbracket_{\alpha^+}$ can be viewed as a restriction of $\llbracket \Gamma \triangleright M \rrbracket_\alpha^U$. The proof is a lengthy induction on term formation.

Theorem 6.2 For any τ -term $\Gamma \triangleright M : \sigma$, any $x \in A^+$, and any $\gamma \in \text{Dom } \theta_\Gamma$,

$$\begin{array}{ccc} A^+ \times \text{Dom } \theta_\Gamma & \xrightarrow{[\![\Gamma \triangleright M]\!]_\alpha^U} & [\![\sigma]\!]_A^U \\ \text{id} \times \theta_\Gamma \downarrow & & \downarrow \theta_\sigma \\ A^+ \times [\![\Gamma]\!]_{A^+} & \xrightarrow{[\![\Gamma \triangleright M]\!]_{\alpha^+}} & [\![\sigma]\!]_{A^+} \end{array}$$

- (1) $[\![\Gamma \triangleright M]\!]_\alpha^U(x, \gamma) \in \text{Dom } \theta_\sigma$, and
- (2) $\theta_\sigma \circ [\![\Gamma \triangleright M]\!]_\alpha^U(x, \gamma) = [\![\Gamma \triangleright M]\!]_{\alpha^+}(x, \theta_\Gamma(\gamma))$. □

The main use of this theorem is in deriving the following fundamental relationship between satisfaction in a coalgebra and in its observable ultrapowers.

Theorem 6.3 (Łoś-type theorem for observable ultrapowers)

If $\Gamma \triangleright \varphi$ is an observable τ -formula, $x \in A^+$ and $(z_1, \dots, z_n) \in [\![\Gamma]\!]_{A^+}$, then $\alpha^+, x, z_1, \dots, z_n \models \Gamma \triangleright \varphi$ if, and only if,

$$\{i \in I : \alpha, f(i), g_1(i), \dots, g_n(i) \models \Gamma \triangleright \varphi\} \in U$$

whenever $x = f^U$ and $(z_1, \dots, z_n) = \theta_\Gamma(g_1^U, \dots, g_n^U)$. □

From this we can conclude that the class of all models of an observable formula is closed under observable ultrapowers:

Corollary 6.4 If $\Gamma \triangleright \varphi$ is observable, then

$$\alpha \models \Gamma \triangleright \varphi \text{ if, and only if, } \alpha^+ \models \Gamma \triangleright \varphi.$$

□

Intrinsic Ultrapowers

A set Φ of ground formulas is *satisfiable* in coalgebra α if there is some state of α at which all members of Φ are true, i.e. some $x \in A$ such that $\alpha, x \models \varphi$ for all $\varphi \in \Phi$. Φ is *finitely satisfiable* if each finite subset of Φ is satisfiable in α . Putting $\varphi^\alpha = \{x \in A : \alpha, x \models \varphi\}$, we see that Φ is finitely satisfiable in α iff the collection $\Phi^\alpha = \{\varphi^\alpha : \varphi \in \Phi\}$ of subsets of A has the finite intersection property.

There is a well-known construction in the theory of ultrapowers that will enable us to force certain finitely α -satisfiable Φ 's become satisfiable in α^+ . By choosing a suitable ultrafilter U it can be arranged that any collection \mathcal{S} of subsets of A with the finite intersection property has a “nonstandard element” in its intersection. This element is an $f^U \in A^U$ such that for each $C \in \mathcal{S}$, $f^U \in_U C$, i.e. $\{i : f(i) \in C\} \in U$.

To see how this is done, let I_A be the set of all finite subsets of the powerset of A . A typical element of I_A is of the form $i = \{C_1, \dots, C_n\}$ with the C_j 's being subsets of A . For each $k \in I_A$, let $I_k = \{i \in I_A : k \subseteq i\}$. The collection $U_A = \{I_k : k \in I_A\}$ has the finite intersection property, since $I_{k_1} \cap \dots \cap I_{k_n}$ contains the element $i = k_1 \cup \dots \cup k_n$. Any ultrafilter U on I_A that extends U_A will be called *intrinsic to A* , and the associated A^U and α^+ will be called *intrinsic ultrapowers*.

Now if \mathcal{S} is a collection of subsets of A with the finite intersection property, let $f : I_A \rightarrow A$ be any function such that $f(i) \in \bigcap (i \cap \mathcal{S})$ whenever $i \cap \mathcal{S} \neq \emptyset$. Note that by the finite intersection property, if $i \cap \mathcal{S} \neq \emptyset$ then $\bigcap (i \cap \mathcal{S}) \neq \emptyset$, so such an f does exist. Then for any $C \in \mathcal{S}$, put $k = \{C\} \in I_A$: if $i \in I_{\{C\}}$ then $C \in i \cap \mathcal{S}$, so $f(i) \in C$. This shows that $I_{\{C\}} \subseteq \{i : f(i) \in C\}$, and so $f^U \in_U C$ as desired.

Theorem 6.5 *Let τ be a type that has at least one non-trivial observable subtype. Suppose that every ground observable τ -formula valid in α is valid also in a τ -coalgebra β . Let α^+ be any intrinsic observable ultrapower of α . Then each state of β is bisimilar to a state of α^+ .*

Proof. Let y be a state of β . If $M : o$ is any ground observable term, let $c_M = \llbracket M \rrbracket_\beta(y) \in \llbracket o \rrbracket$. Let Φ_y be the set of equations $M \approx c_M$ for all ground observable M . By definition, Φ_y is satisfied by y in β . Each finite $\{\varphi_1, \dots, \varphi_n\} \subseteq \Phi_y$ is satisfiable in α , for otherwise the formula

$$\neg(\varphi_1 \wedge \dots \wedge \varphi_n)$$

would be valid in α , hence valid in β by hypothesis, contrary to the fact that this formula is false at y .

Thus the collection Φ_y^α has the finite intersection property. If U is the intrinsic ultrafilter that gives rise to α^+ , then by the above construction there is some $f^U \in A^U$ such that for each M , $f^U \in_U (M \approx c_M)^\alpha$, which means that the set

$$\begin{aligned} I_M &= \{i \in I : \alpha, f(i) \models M \approx c_M\} \\ &= \{i \in I : \llbracket M \rrbracket_\alpha(f(i)) = c_M\} \end{aligned}$$

belongs to U . Since this holds for all ground observable M , f^U is observable by (\dagger) , so $f^U \in A^+$. Also, since $I_M \in U$, Theorem 6.3 gives $\alpha^+, f^U \models M \approx c_M$, so

$$\llbracket M \rrbracket_{\alpha^+}(f^U) = c_M = \llbracket M \rrbracket_\beta(y).$$

Therefore f^U and y assign the same values to all ground observable terms, and so are bisimilar by Theorem 5.8(4). \square

7 Definable Classes of Coalgebras

The tools needed to give a structural characterisation of logically definable classes of coalgebras are now all in place. The following result is the analogue for polynomial functors of Theorem 9.2 of [4] for monomial functors, and the underlying reasoning is the same.

Theorem 7.1 *If τ has at least one non-trivial observable subtype, then for any class K of τ -coalgebras, the following are equivalent.*

- (1) *K is the class of all models of some set of rigid observable formulas.*
- (2) *K is the class of all models of some set of ground observable formulas.*
- (3) *K is closed under disjoint unions, images of bisimulations, and observable ultrapowers.*
- (4) *K is closed under disjoint unions, images of bisimilarity relations, and intrinsic observable ultrapowers.*

Proof. We explain why (4) implies (2), the proofs that (2) implies (1) which implies (3) which implies (4) being either evident or already discussed (Theorem 5.3, Corollary 6.4).

Let Φ be the set of all ground observable formulas that are valid in all members of K . By definition all members of K are models of Φ , so it suffices to prove the converse. Let β be a model of Φ . For each ground observable φ such that $\beta \not\models \varphi$ there must be some $\alpha_\varphi \in K$ such that $\alpha_\varphi \not\models \varphi$ (or else φ belongs to Φ hence $\beta \models \varphi$). Let α be the disjoint union of all these α_φ 's. Then any ground observable formula valid in α is valid in every α_φ , hence valid in β . Therefore if α^+ is an intrinsic observable ultrapower of β , then by Theorem 6.5 the bisimilarity relation from α^+ to β is surjective.

In other words, β is the image under bisimilarity of an intrinsic ultrapower of a disjoint union of coalgebras from K . The closure conditions listed in (4) thus ensure that $\beta \in K$. \square

References

- [1] Aczel, P. and N. Mendler, *A Final Coalgebra Theorem*, in: D. H. Pitt et al., editors, *Category Theory and Computer Science. Proceedings 1989*, Lecture Notes in Computer Science **389**, Springer-Verlag, 1989 pp. 357–365.
- [2] Bell, J. L. and A. B. Slomson, “Models and Ultraproducts,” North-Holland, Amsterdam, 1969.
- [3] Chang, C. C. and H. J. Keisler, “Model Theory,” North-Holland, Amsterdam, 1973.
- [4] Goldblatt, R., *What is the Coalgebraic Analogue of Birkhoff’s Variety Theorem?*, *Theoretical Computer Science*, to appear. Manuscript available at <http://www.mcs.vuw.ac.nz/~rob/papers/what.ps>.

- [5] Hermida, C., “Fibrations, Logical Predicates and Indeterminates,” Ph.D. thesis, University of Edinburgh (1993), techn. rep. LFCS-93-277. Also available as Aarhus Univ. DAIMI Techn. rep. PB-462.
- [6] Hermida, C. and B. Jacobs, *Structural induction and coinduction in a fibrational setting*, Information and Computation **145** (1998), pp. 107–152.
- [7] Jacobs, B., *Objects and Classes, Coalgebraically*, in: B. Freitag, C. B. Jones, C. Lengauer and H.-J. Schek, editors, *Object-Oriented Programming with Parallelism and Persistence*, Kluwer Academic Publishers, 1996 pp. 83–103.
- [8] Jacobs, B., “Categorical Logic and Type Theory,” Elsevier, 1999.
- [9] Jacobs, B., *Exercises in Coalgebraic Specification*, manuscript for the proceedings of the *Mathematics for Information Technology* summer school, Oxford, 2000. Available at <http://www.cs.kun.nl/~bart/PAPERS/>.
- [10] Jacobs, B., *Towards a Duality Result in Coalgebraic Modal Logic*, Electronic Notes in Theoretical Computer Science **33** (2000), <http://www.elsevier.nl/locate/entcs>.
- [11] Kurz, A., *Specifying Coalgebras with Modal Logic*, Electronic Notes in Theoretical Computer Science **11** (1998), <http://www.elsevier.nl/locate/entcs>.
- [12] Moss, L. S., *Coalgebraic Logic*, Annals of Pure and Applied Logic **96** (1999), pp. 277–317.
- [13] Pitts, A. M., *Categorical Logic*, in: S. Abramsky, D. M. Gabbay and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science, Volume 5: Algebraic and Logical Structures*, Oxford University Press, 2000 .
- [14] Reichel, H., *An Approach to Object Semantics Based on Terminal Co-Algebras*, Mathematical Structures in Computer Science **5** (1995), pp. 129–152.
- [15] Röbiger, M., *From Modal Logic to Terminal Coalgebras*, Preprint MATH-AL-3-1998, Technische Universität Dresden (1998), to appear in *Theoretical Computer Science*.
- [16] Rutten, J., *A Calculus of Transition Systems (towards Universal Coalgebra)*, in: A. Ponse, M. de Rijke and Y. Venema, editors, *Modal Logic and Process Algebra*, CSLI Lecture Notes No. 53, CSLI Publications, Stanford, California, 1995 pp. 231–256.
- [17] Rutten, J., *Universal Coalgebra: a Theory of Systems*, Theoretical Computer Science **249** (2000), pp. 3–80.