1

Equational Logic of Polynomial Coalgebras[1]

ROBERT GOLDBLATT

ABSTRACT. Coalgebras of polynomial functors constructed from sets of observable elements have been found useful in modelling various kinds of data types and state-transition systems. This paper presents a calculus of terms for operations on such coalgebras, based on a simple type theory, and develops its semantics. The terms admit a single state-valued *parameter*, but may also have state-valued variables. In a "rigid" term all state-variables are bound.

Boolean combinations of equations between terms of observable type are shown to form a natural language for specifying properties of polynomial coalgebras, and for giving a Hennessy-Milner style logical characterisation of bisimilarity of states: two states are bisimilar when they satisfy the same rigid observable formulas. Also, our syntax of terms is expressive enough to show, alternatively, that two states are bisimilar when they assign the same values to all ground observable terms. The proof involves a characterisation of bisimulations as relations preserved by certain functions induced by "paths" between functors. An analysis of the definability of path actions shows that our language is capable of defining certain modalities. These can be used to express modal assertions to the effect that certain formulas will be true after the execution of a state transition.

1 Introduction and Overview

If $T : \mathbf{Set} \to \mathbf{Set}$ is a functor on the category of sets, then a T-*coalgebra* is a pair (A, α) with A being a set and α a function of the form $A \to TA$. This notion has proven useful in modelling data structures such as lists, streams and trees; transition systems such as automata; and classes in object-oriented programming languages [29, 16, 31, 32]. Generically A is viewed as a set of *states*, and α as a *transition structure*.

In many of the examples just mentioned, the functor T is *polynomial*, i.e. is constructed from constant-valued functors and the identity functor by forming products, exponential functors with constant exponent (which

[1] Prepared using Paul Taylor's `diagrams` package.

we will call *power* functors), and coproducts. In that case (A, α) is a *polynomial coalgebra*. The aim of this paper is to set out an appropriate formal language for specifying properties of polynomial coalgebras and to develop its semantics. A first criterion for "appropriateness" is that the language should provide a logical characterisation, in the style of Hennessy and Milner [13], of the fundamental relation of *bisimilarity* between coalgebraic states. The form of this characterisation is that two states are bisimilar precisely when they satisfy the same formulas of the language under the given semantics. In addition the syntax should provide terms denoting the various operations on coalgebras that are naturally associated with the constructions defining polynomial functors; these operations including projections, pairings, injections, evaluations, lambda abstractions, and functional applications. A consequence of that requirement is that bisimilar states can be alternatively characterised as those assigning the same values to certain terms of "observable" type.

Now relational models of propositional modal logic can be viewed as coalgebras [31] and this has led to a number of proposals of languages with modalities for describing coalgebras [26, 21, 30, 18]. An alternative method used here is to develop a syntax of equations between terms for coalgebraic operations that is analogous to the standard logic of terms for abstract algebras, but subject to the principle that a coalgebraic term should have a single *state*-valued variable or parameter. The result is an equational language that is very similar to those of classical universal algebra and categorical logic, but at the same time is *implicitly modal* in nature. It particular it is able to express modal assertions to the effect that certain formulas will be true after the execution of state transitions (see Theorem 6.5, Corollary 6.6 and Section 8).

A previous article [9] developed such a calculus of terms and equations for polynomial coalgebras that are *monomial*, i.e. constructed without the use of coproducts. It was shown that Boolean combinations of equations between terms of "observable" type form a suitable language of formulas for specifying properties of monomial coalgebras and characterising bisimulation relations between them, both in the Hennessy-Milner style of equivalence of formula satisfaction and in terms of equality of term-evaluation. Our purpose now is to explain how that theory can be extended to include coproducts, whose presence introduces considerable complexity associated with the *partiality* of certain "path functions" expressing the dynamics of the transition structure α.

The approach we take is to use type theory [17] 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. The base types denote fixed sets of observable elements. There is also 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 $\mathsf{tr}(\mathsf{s})$, or more generally $\mathsf{tr}(M)$ for any state-valued term M. But the situation is more subtle than previously, because we now allow state variables distinct from the parameter s in coalgebraic specifications, provided that they are bound. In the syntax of [9] 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 denoting 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.

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 $\iota_j : A_j \to A_1 + A_2$ 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

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

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, giving an element of B as the desired value. Another notation for this term [17, Section 2.3] is

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

Example: To illustrate the use of rigid terms and case-formation in coalgebraic specification, here is an example adapted from [19, 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

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

$$
\begin{array}{l}
\mathit{size}(\mathsf{s}) = \;\mathsf{case}\ \mathit{children}(\mathsf{s})\ \mathsf{of} \\
\qquad\qquad \iota_1 u \mapsto \iota_2 1 \\
\qquad\qquad \iota_2 v \mapsto \mathsf{case}\ \mathit{size}(\pi_1 v)\ \mathsf{of} \\
\qquad\qquad\qquad\qquad \iota_1 u \mapsto \iota_1 * \\
\qquad\qquad\qquad\qquad \iota_2 n \mapsto \mathsf{case}\ \mathit{size}(\pi_2 v)\ \mathsf{of} \\
\qquad\qquad\qquad\qquad\qquad\qquad \iota_1 u \mapsto \iota_1 * \\
\qquad\qquad\qquad\qquad\qquad\qquad \iota_2 k \mapsto \iota_2 (n + k + 1) \\
\qquad\qquad\qquad\qquad\qquad \mathsf{endcase} \\
\qquad\qquad\qquad \mathsf{endcase} \\
\qquad \mathsf{endcase}
\end{array}
$$

Figure 1. case terms

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

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

where $\mathbb{N}$ is the set of positive integers and $\mathit{size}(x) = \iota_1 *$ 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 $\mathit{children} = \pi_1 \circ \alpha$ and $\mathit{size} = \pi_2 \circ \alpha$, 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 $\mathit{size}(\pi_j v)$..., v is bound in M itself. M is rigid.

The notion of a *bisimulation* first appeared in [27] as a relation of mutual simulation between states of two automata. Park showed that if two deterministic automata are related by a bisimulation, then they accept the same set of inputs. Hennessy and Milner [12, 13] introduced the idea of characterising observationally equivalent, or behaviourally indistinguishable, processes as those satisfying the same formulae of a logical language. They provided a simple propositional modal language that achieved this. Observational equivalence was defined as the relation which is in fact the largest bisimulation between processes. Equivalent processes were later dubbed *bisimilar*.

A category-theoretic definition of bisimulation relations between T-coalgebras (A, α) and (B, β) was given in [1]: a relation $R \subseteq A \times B$ is a bisimulation if there is a transition structure $\rho : R \to TR$ such that the projections $A \times B \to A$ and $A \times B \to B$ are coalgebraic morphisms from ρ to α and β (see Section 5). *Bisimilarity* is the largest such relation, which always exists because the union of any collection of bisimulations is a bisimulation. In [15] another approach was introduced by defining a "lifting" of the relation R to a relation $R^T \subseteq TA \times TB$ and taking R to be a bisimulation if it was preserved by the lifting, in the sense that xRy implies $\alpha(x) R^T \beta(y)$. Here we will show that when T is polynomial, the definition can be reformulated with the help of a notion from [18] of a *path* $T \overset{p}{\rightsquigarrow} S$ from T to one of its component functors S. This p is a sequence of symbols that reflects the way S is structured within T. We will see that a path induces certain partial functions $p_X : TX \circ\!\!\longrightarrow SX$ for any set X, and that R is a bisimulation when it is "preserved" by such functions (Theorem 5.7). Moreover, the action of a path function is definable by a term of our language for coalgebras. From this we obtain a logical characterisation of bisimilarity of states as meaning that they assign the same values to all ground terms of observable type, or that they satisfy the same equations between such terms, or indeed that they satisfy the same Boolean combinations of such equations.

To summarize, the main features of this paper are:

- The formulation of syntax and semantics of types and terms for operations in coalgebras of any polynomial functor (Sections 2, 3 and 4).

- The characterisation of bisimulation relations by their preservation under partial functions induced by "paths" between functors, and the term-definability of these path functions (Sections 5 and 6).

- 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 7.2).

This characterisation of bisimilarity is used in an associated article [10] to obtain a structural characterisation of classes of polynomial coalgebras definable by sets of rigid observable formulas. The result is an analogue for polynomial coalgebras of Birkhoff's famous characterisation in [2] of equational classes of abstract algebras as being those closed under direct products, homomorphic images and subalgebras. The coalgebraic analogue requires the development of a new notion of "observational ultrapower" of coalgebras. Yet another approach [7] involves a "Stone space" type of construction on coalgebras, similar to the ultrafilter enlargements used in modal model theory. A brief description of these further developments is provided here in the final Section 9.

Section 8 gives a comparison with other formalisms, including an explanation of why the language developed here subsumes the modal language for polynomial coalgebras of [30].

2 Polynomial Functors

Standard notation for products, powers and coproducts of sets will be used. The *coproduct* $A_1 + A_2$ and associated *insertions* $\iota_j : A_j \to A_1 + A_2$ have already been described. $\pi_j : A_1 \times A_2 \to A_j$ 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 $ev_d : A^D \to A$ having $ev_d(f) = f(d)$. The identity function on a set A is denoted $\mathrm{id}_A : A \to A$.

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 $\mathrm{Dom}\, f \subseteq A$. We may write $f(x)\!\downarrow$ to mean that $f(x)$ is defined, i.e. $x \in \mathrm{Dom}\, f$. Associated with each insertion $\iota_j : A_j \to A_1 + A_2$ 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 $\mathrm{Dom}\, \varepsilon_j = \iota_j A_j$, i.e. $y \in \mathrm{Dom}\, \varepsilon_j$ iff $y = \iota_j(x)$ 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 $T : \mathbf{Set} \to \mathbf{Set}$.

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

$$T_1(f) \times T_2(f) : T_1 A \times T_2 A \to T_1 B \times T_2 B$$

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 $f : A \to B$, has $T_1 + T_2(f)$ being the function

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

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

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

If $\mathrm{Dom}\,\alpha \subseteq \mathrm{Dom}\,\beta$, then α is a *subcoalgebra* of β iff the inclusion function $\mathrm{Dom}\,\alpha \hookrightarrow \mathrm{Dom}\,\beta$ 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 $\mathrm{Dom}\,\alpha_i$'s and whose transition structure acts as α_j on the summand $\iota_j \mathrm{Dom}\,\alpha_j$ of $\mathrm{Dom} \sum_I \alpha_i$. More precisely, this transition is given by $\iota_j(a) \mapsto T(\iota_j)(\alpha_j(a))$, with the insertion $\iota_j : \mathrm{Dom}\,\alpha_j \to \mathrm{Dom} \sum_I \alpha_i$ being an injective morphism making α_j isomorphic to a subcoalgebra of the disjoint union (see [32, Section 4]).

3 Syntax of Types, Terms and Formulas

Types

Fix a set $\mathbb{O}$ of symbols called *observable types*, and a collection $\{[\![o]\!] : o \in \mathbb{O}\}$ of non-empty sets indexed by $\mathbb{O}$. $[\![o]\!]$ is the *denotation* of o, and its members are called *observable elements*, or *constants* of type o.

Example:

$\mathbb{O} = \{\mathsf{num}, \mathsf{bool}, 1\}$, with $[\![\mathsf{num}]\!] = \{0, 1, \dots\}$, $[\![\mathsf{bool}]\!] = \{\mathit{true}, \mathit{false}\}$, $[\![1]\!] = \{*\}$.

The set of *types over* $\mathbb{O}$, or $\mathbb{O}$*-types*, is the smallest set $\mathbb{T}$ such that $\mathbb{O} \subseteq \mathbb{T}$, $\mathsf{St} \in \mathbb{T}$ 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. Thus the set $\mathit{Sub}\,\tau$ of subtypes of τ is just $\{\tau\}$ when $\tau \in \mathbb{O} \cup \{\mathsf{St}\}$; is equal to $\{\tau\} \cup \mathit{Sub}\,\sigma_1 \cup \mathit{Sub}\,\sigma_2$ when τ is $\sigma_1 \times \sigma_2$ or $\sigma_1 + \sigma_2$; and is $\{\tau\} \cup \{o\} \cup \mathit{Sub}\,\sigma$ when τ is $o \Rightarrow \sigma$.

St is a type symbol that will denote the state set of a given coalgebra. 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.

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).

Each $\mathbb{O}$-type σ determines a polynomial functor $|\sigma| : \mathbf{Set} \to \mathbf{Set}$. For $o \in \mathbb{O}$, $|o|$ is the constant functor $\bar{D}$ where $D = [\![\, o \,]\!]$; $|\mathsf{St}|$ is the identity functor Id; and inductively

$$|\sigma_1 \times \sigma_2| = |\sigma_1| \times |\sigma_2|, \quad |\sigma_1 + \sigma_2| = |\sigma_1| + |\sigma_2|, \quad |o \Rightarrow \sigma| = |\sigma|^{[\![\, o \,]\!]}.$$

For denotations of types, we write $[\![\, \sigma \,]\!]_A$ for the set $|\sigma|A$. Thus we have $[\![\, o \,]\!]_A = [\![\, o \,]\!]$, $[\![\, \mathsf{St} \,]\!]_A = A$,

$$\begin{aligned} [\![\, \sigma_1 \times \sigma_2 \,]\!]_A &= [\![\, \sigma_1 \,]\!]_A \times [\![\, \sigma_2 \,]\!]_A \\ [\![\, \sigma_1 + \sigma_2 \,]\!]_A &= [\![\, \sigma_1 \,]\!]_A + [\![\, \sigma_2 \,]\!]_A \\ [\![\, o \Rightarrow \sigma \,]\!]_A &= [\![\, \sigma \,]\!]_A{}^{[\![\, o \,]\!]}. \end{aligned}$$

If σ is a rigid type then $|\sigma|$ is a constant functor, so $[\![\, \sigma \,]\!]_A$ is a fixed set whose definition does not depend on A and may be written $[\![\, \sigma \,]\!]$. A τ*-coalgebra* is a coalgebra (A, α) for the functor $|\tau|$, i.e. α is a function of the form $A \to [\![\, \tau \,]\!]_A$.

Terms

To define *terms* we fix a denumerable set *Var* of *variables* and define a *context* to be a finite (possible 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 $v_1, \ldots, v_n$ 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

$$\Gamma \rhd M : \sigma,$$

which signifies that M is a "raw" term of type σ in context Γ. This may be abbreviated to $\Gamma \rhd M$ if the type of the term is understood. If $\sigma \in \mathbb{O}$, then the term is *observable.*

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

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 $\Gamma \rhd M$, all free variables of M appear in the list Γ. A *ground* term is one of the form $\emptyset \rhd M : \sigma$, which may be abbreviated to $M : \sigma$, or just to the raw term M. Thus a ground term has no free variables. Note that a ground term may contain the state parameter s, which behaves nonetheless as a variable in that it can denote any member of $\mathrm{Dom}\,\alpha$, as will be seen in the semantics presented in Section 4.

LEMMA 3.1 *There exist ground terms of every type.*

Proof. (Con) provides ground terms of all observable types, since $[\![o]\!] \neq \emptyset$ when $o \in \mathbb{O}$, and (St) provides a term of type St. The rules (Pair), (In_j) and (Abs), with the help of (Weak), then allow the inductive formation of ground terms of all other types. ■

A term is defined to be *rigid* if its context is rigid. This entails that any free variable of the term is assigned a rigid type by Γ, so its type is formed without use of St. Of course all ground terms are rigid.

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

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

Axioms

(Var) $$\frac{v \in Var}{v : \sigma \rhd v : \sigma}$$ (Con) $$\frac{c \in [\![o]\!]}{\emptyset \rhd c : o}$$ (St) $$\frac{}{\emptyset \rhd \mathsf{s} : \mathsf{St}}$$

Weakening

(Weak) $$\frac{\Gamma, \Gamma' \rhd M : \sigma}{\Gamma, v : \sigma', \Gamma' \rhd M : \sigma}$$ where v does not occur in Γ or Γ'.

Product Types

(Pair) $$\frac{\Gamma \rhd M_1 : \sigma_1 \qquad \Gamma \rhd M_2 : \sigma_2}{\Gamma \rhd \langle M_1, M_2 \rangle : \sigma_1 \times \sigma_2}$$

(Proj$_1$) $$\frac{\Gamma \rhd M : \sigma_1 \times \sigma_2}{\Gamma \rhd \pi_1 M : \sigma_1}$$ (Proj$_2$) $$\frac{\Gamma \rhd M : \sigma_1 \times \sigma_2}{\Gamma \rhd \pi_2 M : \sigma_2}$$

Coproduct Types

(In$_1$) $$\frac{\Gamma \rhd M : \sigma_1}{\Gamma \rhd \iota_1 M : \sigma_1 + \sigma_2}$$ (In$_2$) $$\frac{\Gamma \rhd M : \sigma_2}{\Gamma \rhd \iota_2 M : \sigma_1 + \sigma_2}$$

(Case) $$\frac{\Gamma \rhd N : \sigma_1 + \sigma_2 \qquad \Gamma, v_1 : \sigma_1 \rhd M_1 : \sigma \qquad \Gamma, v_2 : \sigma_2 \rhd M_2 : \sigma}{\Gamma \rhd \mathsf{case}\ N\ \mathsf{of}\ [\iota_1 v_1 \mapsto M_1 \mid \iota_2 v_2 \mapsto M_2] : \sigma}$$

Power Types

(Abs) $$\frac{\Gamma, v : o \rhd M : \sigma}{\Gamma \rhd (\lambda v. M) : o \Rightarrow \sigma}$$ (App) $$\frac{\Gamma \rhd M : o \Rightarrow \sigma \qquad \Gamma \rhd N : o}{\Gamma \rhd M \cdot N : \sigma}$$

Figure 2. Axioms and Rules for Generating Terms

Equations

$$\text{(Eq)}\quad \frac{\Gamma \rhd M_1 : \sigma \qquad \Gamma \rhd M_2 : \sigma}{\Gamma \rhd M_1 \approx M_2}$$

Weakening

$$\text{(Weak)}\quad \frac{\Gamma, \Gamma' \rhd \varphi}{\Gamma, v : \sigma', \Gamma' \rhd \varphi} \qquad \text{where } v \text{ does not occur in } \Gamma \text{ or } \Gamma'.$$

Connectives

$$\text{(Neg)}\quad \frac{\Gamma \rhd \varphi}{\Gamma \rhd \neg\varphi} \qquad\qquad \text{(Con)}\quad \frac{\Gamma \rhd \varphi_1 \qquad \Gamma \rhd \varphi_2}{\Gamma \rhd \varphi_1 \wedge \varphi_2}$$

Figure 3. Formation Rules for Formulas

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

$$\emptyset \rhd \mathsf{tr}(\mathsf{s}) : \tau.$$

The symbol tr denotes the transition structure of a τ-coalgebra $A \xrightarrow{\alpha} [\![\tau]\!]_A$. If M is interpreted as the state x of α, then $\mathsf{tr}(M)$ is interpreted as $\alpha(x)$.

τ-Formulas

An *equation-in-context* has the form $\Gamma \rhd M_1 \approx M_2$ where $\Gamma \rhd M_1$ and $\Gamma \rhd M_2$ are terms *of the same type*. A *formula-in-context* has the form $\Gamma \rhd \varphi$, with the raw 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 $\emptyset \rhd \varphi$ with empty context is *ground*, and may be abbreviated to 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

Types σ and contexts $\Gamma = (v_1 : \sigma_1, \ldots, v_n : \sigma_n)$ are interpreted in a given τ-coalgebra $\alpha : A \to |\tau|A$ by putting $[\![\sigma]\!]_\alpha = |\sigma|(\mathrm{Dom}\,\alpha) = [\![\sigma]\!]_A$, and

$$[\![\Gamma]\!]_\alpha = [\![\sigma_1]\!]_\alpha \times \cdots \times [\![\sigma_n]\!]_\alpha$$

(so $[\![\,\emptyset\,]\!]_\alpha$ is the empty product 1). The *denotation* of each τ-term $\Gamma \rhd M : \sigma$, relative to the coalgebra α, is a function

$$[\![\,\Gamma \rhd M : \sigma\,]\!]_\alpha : A \times [\![\,\Gamma\,]\!]_\alpha \longrightarrow [\![\,\sigma\,]\!]_\alpha,$$

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

$$A \times [\![\,\emptyset\,]\!]_\alpha = A \times 1 \cong A,$$

so we replace $A \times [\![\,\emptyset\,]\!]_\alpha$ by A itself and interpret a ground term $\emptyset \rhd M : \sigma$ as a function $A \to [\![\,\sigma\,]\!]_\alpha$. The definition of denotations is as follows.

Var:

$[\![\,v : \sigma \rhd v : \sigma\,]\!]_\alpha : A \times [\![\,\sigma\,]\!]_\alpha \to [\![\,\sigma\,]\!]_\alpha$ is the right projection function.

Con:

$[\![\,\emptyset \rhd c : o\,]\!]_\alpha : A \to [\![\,o\,]\!]$ is the constant function with value c.

St:

$[\![\,\emptyset \rhd \mathsf{s} : \mathsf{St}\,]\!]_\alpha : A \to [\![\,\mathsf{St}\,]\!]_\alpha$ is the identity function $A \to A$.

τ-Tr:

$[\![\,\Gamma \rhd \mathsf{tr}(M) : \tau\,]\!]_\alpha : A \times [\![\,\Gamma\,]\!]_\alpha \to [\![\,\tau\,]\!]_\alpha$ is the composition of the functions

$$A \times [\![\,\Gamma\,]\!]_\alpha \xrightarrow{[\![\,\Gamma \rhd M : \mathsf{St}\,]\!]_\alpha} A \xrightarrow{\ \alpha\ } [\![\,\tau\,]\!]_\alpha.$$

Weak:

$[\![\,\Gamma, v : \sigma', \Gamma' \rhd M : \sigma\,]\!]_\alpha$ is the composition of $[\![\,\Gamma, \Gamma' \rhd M : \sigma\,]\!]_\alpha$ with the projection

$$A \times [\![\,\Gamma\,]\!]_\alpha \times [\![\,\sigma'\,]\!]_\alpha \times [\![\,\Gamma'\,]\!]_\alpha \longrightarrow A \times [\![\,\Gamma\,]\!]_\alpha \times [\![\,\Gamma'\,]\!]_\alpha.$$

Pair:

$[\![\,\Gamma \rhd \langle M_1, M_2\rangle : \sigma_1 \times \sigma_2\,]\!]_\alpha$ is the product map

$$A \times [\![\,\Gamma\,]\!]_\alpha \xrightarrow{\langle\, [\![\,\Gamma \rhd M_1 : \sigma_1\,]\!]_\alpha,\, [\![\,\Gamma \rhd M_2 : \sigma_2\,]\!]_\alpha\,\rangle} [\![\,\sigma_1\,]\!]_\alpha \times [\![\,\sigma_2\,]\!]_\alpha.$$

Proj$_j$:

$[\![\,\Gamma \rhd \pi_j M : \sigma_j\,]\!]_\alpha$ is the composition of

$$A \times [\![\,\Gamma\,]\!]_\alpha \xrightarrow{[\![\,\Gamma \rhd M : \sigma_1 \times \sigma_2\,]\!]_\alpha} [\![\,\sigma_1\,]\!]_\alpha \times [\![\,\sigma_2\,]\!]_\alpha \xrightarrow{\ \pi_j\ } [\![\,\sigma_j\,]\!]_\alpha.$$

Inj$_j$:

$[\![\, \Gamma \rhd \iota_j M : \sigma_1 + \sigma_2 \,]\!]_\alpha$ is the composition of

$$A \times [\![\, \Gamma \,]\!]_\alpha \xrightarrow{[\![\, \Gamma \rhd M : \sigma_j \,]\!]_\alpha} [\![\, \sigma_j \,]\!]_\alpha \xrightarrow{\iota_j} [\![\, \sigma_1 \,]\!]_\alpha + [\![\, \sigma_2 \,]\!]_\alpha .$$

Case:

This is easier to describe at the function-value level. For $x \in A$ and $\gamma \in [\![\, \Gamma \,]\!]_\alpha$, let

$$[\![\, \Gamma \rhd N : \sigma_1 + \sigma_2 \,]\!]_\alpha (x, \gamma) = \iota_j(a) \in [\![\, \sigma_1 \,]\!]_\alpha + [\![\, \sigma_2 \,]\!]_\alpha$$

(which holds for a unique j and $a \in [\![\, \sigma_j \,]\!]_\alpha$). Then the element

$$[\![\, \Gamma \rhd \mathsf{case}\ N\ \mathsf{of}\ [\iota_1 v_1 \mapsto M_1 \mid \iota_2 v_2 \mapsto M_2] : \sigma \,]\!]_\alpha (x, \gamma)$$

of $[\![\, \sigma \,]\!]_\alpha$ is defined to be

$$[\![\, \Gamma, v_j : \sigma_j \rhd M_j : \sigma \,]\!]_\alpha (x, \gamma, a).$$

Put more succinctly: if $[\![\, \Gamma \rhd N \,]\!]_\alpha (x, \gamma) \in \mathrm{Dom}\, \varepsilon_j$, then

$$[\![\, \Gamma \rhd \mathsf{case}\ N\ \mathsf{of} .. M_1 | .. M_2 \,]\!]_\alpha (x, \gamma) = [\![\, \Gamma, v_j : \sigma_j \rhd M_j \,]\!]_\alpha (x, \gamma, \varepsilon_j [\![\, \Gamma \rhd N \,]\!]_\alpha (x, \gamma)).$$

Abs:

$[\![\, \Gamma \rhd (\lambda v.M) : o \Rightarrow \sigma \,]\!]_\alpha (x, \gamma)$ is the function $[\![\, o \,]\!] \to [\![\, \sigma \,]\!]_\alpha$ given by

$$a \mapsto [\![\, \Gamma, v : o \rhd M : \sigma \,]\!]_\alpha (x, \gamma, a).$$

App:

$[\![\, \Gamma \rhd M \cdot N : \sigma \,]\!]_\alpha (x, \gamma)$ is the element of $[\![\, \sigma \,]\!]_\alpha$ obtained by evaluating the function

$$[\![\, \Gamma \rhd M : o \Rightarrow \sigma \,]\!]_\alpha (x, \gamma) : [\![\, o \,]\!] \longrightarrow [\![\, \sigma \,]\!]_\alpha$$

at $[\![\, \Gamma \rhd N : o \,]\!]_\alpha (x, \gamma) \in [\![\, o \,]\!]$.

This completes the inductive definition of $[\![\, \Gamma \rhd M : \sigma \,]\!]_\alpha$.

An important observation for what follows is that if $\Gamma \rhd M : \sigma$ is "tr-free", meaning that the symbol tr does not occur in the raw term M, then the denotation $[\![\, \Gamma \rhd M : \sigma \,]\!]_\alpha$ depends only on A, not on α, and may be written $[\![\, \Gamma \rhd M : \sigma \,]\!]_A$. This is readily shown by induction on the formation of $\Gamma \rhd M : \sigma$.

Substitution of Terms

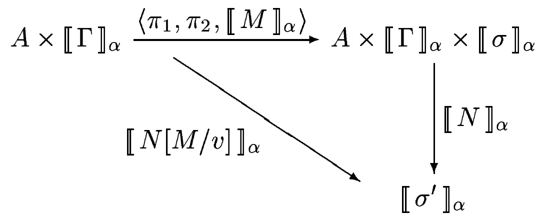
In working with this system it becomes essential to have available the operation $N[M/v]$ of substituting the raw term M for free occurrences of the variable v in N. The following rule is derivable:

$$\text{(Subst)} \quad \frac{\Gamma \rhd M : \sigma \qquad \Gamma, v : \sigma \rhd N : \sigma'}{\Gamma \rhd N[M/v] : \sigma'}$$

The semantics of terms obeys the basic principle that substitution is interpreted as *composition* of denotations [28, 2.2]. Because of the special role of the state set A, this takes the form

$$[\![\, \Gamma \rhd N[M/v] \,]\!]_\alpha = [\![\, \Gamma, v : \sigma \rhd N \,]\!]_\alpha \circ \langle \pi_1, \pi_2, [\![\, \Gamma \rhd M \,]\!]_\alpha \rangle,$$

so that the following diagram commutes:



Substitution of a raw term can also take place in a *formula*, producing expressions $\varphi[M/v]$ where φ is a raw formula. Thus $(M_1 \approx M_2)[N/v]$ is $(M_1[N/v] \approx M_2[N/v])$ etc. With the help of (Subst), the following rule can be derived:

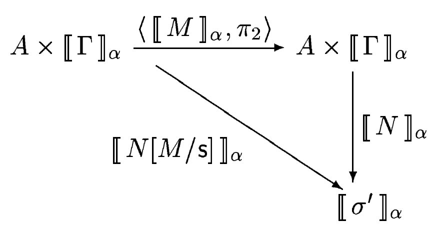
$$\frac{\Gamma \rhd M : \sigma \qquad \Gamma, v : \sigma \rhd \varphi}{\Gamma \rhd \varphi[M/v]}$$

Substitution for the State Parameter

It is also possible to make substitutions $N[M/\mathsf{s}]$ for the state parameter s according to the derivable rule

$$\text{(s-Subst)} \quad \frac{\Gamma \rhd M : \mathsf{St} \qquad \Gamma \rhd N : \sigma'}{\Gamma \rhd N[M/\mathsf{s}] : \sigma'}$$

with the semantics $[\![\, \Gamma \rhd N[M/\mathsf{s}] \,]\!]_\alpha = [\![\, \Gamma \rhd N \,]\!]_\alpha \circ \langle\, [\![\, \Gamma \rhd M \,]\!]_\alpha, \pi_2 \rangle$:



In the case of ground terms ($\Gamma = \emptyset$), this simplifies to

$$[\![N[M/\mathsf{s}]]\!]_\alpha = [\![N]\!]_\alpha \circ [\![M]\!]_\alpha,$$

an equation that will play a significant role below.

Terms for Transitions

In any τ-coalgebra (A, α), the term function $[\![\mathsf{tr}(\mathsf{s}) : \tau]\!]_\alpha$ is just α itself. If $\tau = \sigma_1 \times \sigma_2$, then $\pi_j \circ \alpha : A \to [\![\sigma_j]\!]_A$ is a σ_j-coalgebra, and a simple calculation shows that

$$[\![\mathsf{tr}(\mathsf{s}) : \sigma_j]\!]_{\pi_j \circ \alpha} = \pi_j \circ [\![\mathsf{tr}(\mathsf{s}) : \sigma_1 \times \sigma_2]\!]_\alpha = [\![\pi_j \mathsf{tr}(\mathsf{s}) : \sigma_j]\!]_\alpha. \tag{4.1}$$

Similarly if τ is $o \Rightarrow \sigma$, then for each $d \in [\![o]\!]$, $ev_d \circ \alpha : A \to [\![\sigma]\!]_A$ is a σ-coalgebra with

$$[\![\mathsf{tr}(\mathsf{s}) : \sigma]\!]_{ev_d \circ \alpha} = [\![\mathsf{tr}(\mathsf{s}) \cdot d : \sigma]\!]_\alpha. \tag{4.2}$$

Term-Definability of Extractions

To obtain a result for extractions similar to equations (4.1) and (4.2), we first show that the action of an extraction function ε_j is term-definable:

LEMMA 4.1 *For any term $\Gamma \rhd M : \sigma_1 + \sigma_2$ of coproduct type there exist terms $\Gamma \rhd \varepsilon_j M : \sigma_j$ for $j = 1, 2$ such that whenever $[\![\Gamma \rhd M]\!]_\alpha(x, \gamma) \in \iota_j [\![\sigma_j]\!]_A$ then*

$$[\![\Gamma \rhd \varepsilon_j M]\!]_\alpha(x, \gamma) = \varepsilon_j \circ [\![\Gamma \rhd M]\!]_\alpha(x, \gamma) \in [\![\sigma_j]\!]_A.$$

Proof. Letting v_1, v_2 be variables not occurring in M, put

$$\varepsilon_1 M := \mathsf{case}\ M\ \mathsf{of}\ [\iota_1 v_1 \mapsto v_1 \mid \iota_2 v_2 \mapsto N_2],$$

where $\emptyset \rhd N_2 : \sigma_1$ is any ground term of type σ_1 (Lemma 3.1). Using (Var) and (Weak) we derive terms $\Gamma, v_1 : \sigma_1 \rhd v_1 : \sigma_1$ and $\Gamma \rhd N_2 : \sigma_1$, from which the (Case) formation rule yields $\Gamma \rhd \varepsilon_1 M : \sigma_1$. Then if $[\![\Gamma \rhd M]\!]_\alpha(x, \gamma) = \iota_1(a)$, the definition of the denotation of case gives

$$[\![\Gamma \rhd \varepsilon_1 M]\!]_\alpha(x, \gamma) = [\![\Gamma, v_1 : \sigma_1 \rhd v_1 : \sigma_1]\!]_\alpha(x, \gamma, a) = a = \varepsilon_1([\![\Gamma \rhd M]\!]_\alpha(x, \gamma))$$

as desired. Similarly, $\varepsilon_2 M := \mathsf{case}\ M\ \mathsf{of}\ [\iota_1 v_1 \mapsto N_1 \mid \iota_2 v_2 \mapsto v_2]$ with $\emptyset \rhd N_1 : \sigma_2$.

■

The "extraction terms" terms $\varepsilon_j M$ will play an important role later. Here we note that if (A, α) is a $\sigma_1 + \sigma_2$-coalgebra, and $\alpha_j : A \to [\![\sigma_j]\!]_A$ is an extension of the partial function $\varepsilon_j \circ \alpha$, then $\alpha(x) \in \iota_j [\![\sigma_j]\!]_A$ implies $\alpha_j(x) = \varepsilon_j(\alpha(x))$ and so by Lemma 4.1,

$$[\![\mathsf{tr}(\mathsf{s}) : \sigma_j]\!]_{\alpha_j}(x) = \varepsilon_j([\![\mathsf{tr}(\mathsf{s}) : \sigma_1 + \sigma_2]\!]_\alpha(x)) = [\![\varepsilon_j \mathsf{tr}(\mathsf{s}) : \sigma_j]\!]_\alpha(x). \tag{4.3}$$

Also we will need a technical fact about the behaviour of $\varepsilon_j M$ under substitution in the case that M is a variable v: it is readily seen that

$$\varepsilon_j v[\mathsf{tr}(\mathsf{s})/v] = \varepsilon_j \mathsf{tr}(\mathsf{s}). \tag{4.4}$$

Semantics of Formulas

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

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

for τ-formulas in τ-coalgebras, which expresses that $\Gamma \rhd \varphi$ is *satisfied*, or *true*, in α at state x under the value-assigment $\gamma \in [\![\Gamma]\!]_\alpha$ to the variables of context Γ. This is defined inductively by

$$\begin{array}{lll} \alpha, x, \gamma \models \Gamma \rhd M_1 \approx M_2 & \text{iff} & [\![\Gamma \rhd M_1]\!]_\alpha(x, \gamma) = [\![\Gamma \rhd M_2]\!]_\alpha(x, \gamma), \\ \alpha, x, \gamma \models \Gamma \rhd \neg\varphi & \text{iff} & \text{not } \alpha, x, \gamma \models \Gamma \rhd \varphi, \\ \alpha, x, \gamma \models \Gamma \rhd \varphi_1 \wedge \varphi_2 & \text{iff} & \alpha, x, \gamma \models \Gamma \rhd \varphi_1 \text{ and } \alpha, x, \gamma \models \Gamma \rhd \varphi_2. \end{array}$$

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

Semantics Under Transitions

Results (4.1)–(4.3) give rise to corresponding results about satisfaction of certain formulas in a coalgebra α and in derivative coalgebras based on the same state set.

Suppose that a formula $(v : \tau \rhd \varphi)$ is tr-free. Thus any raw term M in φ has no occurrence of tr, so in any τ-coalgebra (A, α) the term-denotation $[\![v : \tau \rhd M]\!]_\alpha$ is independent of the transition α and therefore may be written $[\![v : \tau \rhd M]\!]_A$. If $\tau = \sigma_1 \times \sigma_2$, then in the σ_j-coalgebra $\pi_j \circ \alpha : A \to [\![\sigma_j]\!]_A$ we find that in general

$$\pi_j \circ \alpha, x \models \varphi[\mathsf{tr}(\mathsf{s})/v] \quad \text{iff} \quad \alpha, x \models \varphi[\pi_j \mathsf{tr}(\mathsf{s})/v]. \tag{4.5}$$

To see this, suppose that φ is the equation $M_1 \approx M_2$, so $\varphi[\mathsf{tr}(\mathsf{s})/v]$ is the equation $M_1[\mathsf{tr}(\mathsf{s})/v] \approx M_2[\mathsf{tr}(\mathsf{s})/v]$, and likewise for $\varphi[\pi_j \mathsf{tr}(\mathsf{s})/v]$. But for $i = 1, 2$,

$$
\begin{array}{ll}
[\![M_i[\mathsf{tr}(\mathsf{s})/v]]\!]_{\pi_j \circ \alpha}(x) & \\
= [\![v : \tau \rhd M_i]\!]_{\pi_j \circ \alpha}(x, [\![\mathsf{tr}(\mathsf{s}) : \sigma_j]\!]_{\pi_j \circ \alpha}(x)) & \text{semantics of (Subst)} \\
= [\![v : \tau \rhd M_i]\!]_{A}(x, [\![\pi_j \mathsf{tr}(\mathsf{s}) : \sigma_j]\!]_{\alpha}(x)) & \text{as } M_i \text{ is } \mathsf{tr}\text{-free, and (4.1)} \\
= [\![v : \tau \rhd M_i]\!]_{\alpha}(x, [\![\pi_j \mathsf{tr}(\mathsf{s}) : \sigma_j]\!]_{\alpha}(x)) & \text{as } M_i \text{ is } \mathsf{tr}\text{-free} \\
= [\![M_i[\pi_j \mathsf{tr}(\mathsf{s})/v]]\!]_{\alpha}(x), &
\end{array}
$$

from which (4.5) readily follows in this case. The inductive cases for formulas $\neg\varphi$ and $\varphi_1 \wedge \varphi_2$ are straightforward.

Similarly if τ is $o \Rightarrow \sigma$, then with the help of (4.2) we can show that for each $d \in [\![o]\!]$, the σ-coalgebra $ev_d \circ \alpha : A \to [\![\sigma]\!]_A$ has

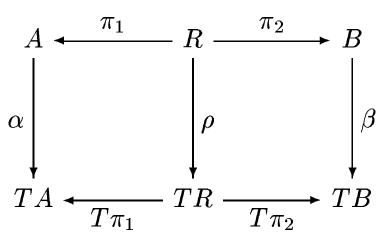
$$ev_d \circ \alpha, x \models \varphi[\mathsf{tr}(\mathsf{s})/v] \quad \text{iff} \quad \alpha, x \models \varphi[\mathsf{tr}(\mathsf{s}) \cdot d/v]. \tag{4.6}$$

Finally there is the case $\tau = \sigma_1 + \sigma_2$. If $\alpha_j : A \to [\![\sigma_j]\!]_A$ is an extension of the partial function $\varepsilon_j \circ \alpha$, then using (4.3), if $\alpha(x) \in \iota_j[\![\sigma_j]\!]_A$,

$$\alpha_j, x \models \varphi[\mathsf{tr}(\mathsf{s})/v] \quad \text{iff} \quad \alpha, x \models \varphi[\varepsilon_j \mathsf{tr}(\mathsf{s})/v]. \tag{4.7}$$

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 $\rho : R \to TR$ 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:



We may say that coalgebra β is the *image* of the bisimulation, or is the image of α under the bisimulation, if R is surjective, i.e. every member of B is in the image of R. Dually, α is the *domain* of the bisimulation if R is a total relation, i.e. $\mathrm{Dom}\, R = A$.

A function $f : A \to B$ is a morphism from α to β iff its graph $\{(a, f(a)) : a \in A\}$ is a bisimulation from α to β [32, Theorem 2.5]: a morphism is essentially a functional bisimulation. When $\mathrm{Dom}\,\alpha \subseteq \mathrm{Dom}\,\beta$, α is a subcoalgebra of β iff the identity relation on $\mathrm{Dom}\,\alpha$ 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 [14, 15]. 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:

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

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

THEOREM 5.1 *If $R \subseteq \mathrm{Dom}\,\alpha \times \mathrm{Dom}\,\beta$, where α and β are T-coalgebras, then R is a bisimulation from α to β if, and only if, for all states x in α and y in β,*

$$xRy \text{ implies } \alpha(x) R^T \beta(y). \qquad \blacksquare$$

The inverse of a bisimulation is a bisimulation, and the union of any collection of bisimulations from α to β is a bisimulation [32, 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 will now be used to show that bisimulation relations are stable under the action of term denotations, 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 $R^{|\sigma|} \subseteq [\![\sigma]\!]_A \times [\![\sigma]\!]_B$, where $|\sigma|$ is the functor defined by σ. For observable σ, $R^{|\sigma|}$ is just the identity relation on $[\![\sigma]\!]$. 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 $R^\Gamma \subseteq [\![\Gamma]\!]_A \times [\![\Gamma]\!]_B$ to be 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 $[\![\Gamma]\!] = [\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!]$.

THEOREM 5.2 *Suppose that R is a $|\tau|$-bisimulation from α to β, and let $\Gamma \rhd M : \sigma$ be any τ-term.*

(1) *If* $\gamma \in [\![\Gamma]\!]_\alpha$ *and* $\gamma' \in [\![\Gamma]\!]_\beta$ *have* $\gamma R^\Gamma \gamma'$, *then*

$$xRy \text{ implies } [\![\Gamma \rhd M]\!]_\alpha(x,\gamma)\, R^{|\sigma|}\, [\![\Gamma \rhd M]\!]_\beta(y,\gamma').$$

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

$$xRy \text{ implies } [\![\Gamma \rhd M]\!]_\alpha(x,\gamma) = [\![\Gamma \rhd M]\!]_\beta(y,\gamma').$$

(3) *If* $\Gamma \rhd M$ *is a rigid term of observable type, and* $\gamma \in [\![\Gamma]\!]$, *then*

$$xRy \text{ implies } [\![\Gamma \rhd M]\!]_\alpha(x,\gamma) = [\![\Gamma \rhd M]\!]_\beta(y,\gamma).$$

Proof. (3) is a special case of (2), which is a special case of (1). The latter is proven by induction on the formation of the term $\Gamma \rhd M : \sigma$. To consider first the cases of base terms, suppose that $\gamma R^\Gamma \gamma'$ and xRy.

If $\Gamma \rhd M : \sigma$ is $v : \sigma \rhd v : \sigma$, then $[\![\Gamma \rhd M]\!]_\alpha(x,\gamma) = \gamma$ and $[\![\Gamma \rhd M]\!]_\beta(y,\gamma') = \gamma'$. But $\gamma R^{|\sigma|}\gamma'$ because $\gamma R^\Gamma \gamma'$ and $R^\Gamma = R^{|\sigma|}$ (so this case does not depend on whether xRy).

If $\Gamma \rhd M : \sigma$ is $\emptyset \rhd c : o$, then $[\![\Gamma \rhd M]\!]_\alpha(x) = [\![\Gamma \rhd M]\!]_\beta(y) = c$, and $cR^{|\sigma|}c$ as $R^{|\sigma|}$ is the identity relation on $[\![o]\!]$.

If $\Gamma \rhd M : \sigma$ is $\emptyset \rhd \mathsf{s} : \mathsf{St}$, then $[\![\Gamma \rhd M]\!]_\alpha(x) = x$ and $[\![\Gamma \rhd M]\!]_\beta(y) = y$. But $xR^{|\sigma|}y$ because $R^{|\sigma|} = R^{|\mathsf{St}|} = R$ and xRy.

Now suppose $\Gamma \rhd M : \sigma$ is $\Gamma \rhd \mathsf{tr}(M) : \tau$, derived by rule ($\tau$-Tr), and assume that result (1) holds for $\Gamma \rhd M : \mathsf{St}$. If $\gamma R^\Gamma \gamma'$ and xRy, then it follows from this assumption that $[\![\Gamma \rhd M]\!]_\alpha(x,\gamma)\, R\, [\![\Gamma \rhd M]\!]_\beta(y,\gamma')$. But here R is a $|\sigma|$-bisimulation, so Theorem 5.1 applies to yield $\alpha([\![\Gamma \rhd M]\!]_\alpha(x,\gamma))\, R^{|\sigma|}\, \beta([\![\Gamma \rhd M]\!]_\beta(y,\gamma'))$. Hence $[\![\Gamma \rhd \mathsf{tr}(M)]\!]_\alpha(x,\gamma)\, R^{|\sigma|}\, [\![\Gamma \rhd \mathsf{tr}(M)]\!]_\beta(y,\gamma')$, so result (1) holds for $\Gamma \rhd \mathsf{tr}(M) : \tau$.

Next suppose $\sigma = \sigma_1 + \sigma_2$ and, for $j = 1, 2$ make the assumption that (1) holds for the term $\Gamma \rhd M : \sigma_j$. If $\gamma R^\Gamma \gamma'$ and xRy, let $a = [\![\Gamma \rhd M]\!]_\alpha(x,\gamma)$ and $b = [\![\Gamma \rhd M]\!]_\beta(y,\gamma')$. Then by hypothesis $aR^{|\sigma_j|}b$, and so $\iota_j(a)R^{|\sigma|}\iota_j(b)$. But $\iota_j(a) = [\![\Gamma \rhd \iota_j M]\!]_\alpha(x,\gamma)$ and $\iota_j(b) = [\![\Gamma \rhd \iota_j M]\!]_\beta(x,\gamma')$, so (1) holds for $\Gamma \rhd \iota_j M : \sigma_1 + \sigma_2$.

Now let M be the raw term ($\mathsf{case}\ N\ \mathsf{of}\ [\iota_1 v_1 \mapsto M_1 \mid \iota_2 v_2 \mapsto M_2]$), and suppose $\Gamma \rhd M : \sigma$ is derived by the rule (Case). Assume that (1) holds for the premisses $\Gamma \rhd N : \sigma_1 + \sigma_2$ and $\Gamma, v_j : \sigma_j \rhd M_j : \sigma$ of that rule. If $\gamma R^\Gamma \gamma'$ and xRy, let $a = [\![\Gamma \rhd N]\!]_\alpha(x,\gamma)$ and $b = [\![\Gamma \rhd N]\!]_\beta(y,\gamma')$. Then by result (1) for $\Gamma \rhd N$, $aR^{|\sigma_1+\sigma_2|}b$, so for some j, $a \in \iota_j[\![\sigma_j]\!]_A$ and $b \in \iota_j[\![\sigma_j]\!]_B$ and $\varepsilon_j(a)R^{|\sigma_j|}\varepsilon_j(b)$. But then, by result (1) for $\Gamma, v_j : \sigma_j \rhd M_j : \sigma$,

$$[\![\Gamma, v_j : \sigma_j \rhd M_j]\!]_\alpha(x,\gamma,\varepsilon_j(a))\, R^{|\sigma|}\, [\![\Gamma, v_j : \sigma_j \rhd M_j]\!]_\beta(y,\gamma',\varepsilon_j(b)).$$

Since the semantics of case-terms tells us that

$$\llbracket \Gamma \rhd M \rrbracket_\alpha(x,\gamma) = \llbracket \Gamma, v_j : \sigma_j \rhd M_j \rrbracket_\alpha(x,\gamma,\varepsilon_j(a)), \quad \text{and}$$
$$\llbracket \Gamma \rhd M \rrbracket_\beta(y,\gamma') = \llbracket \Gamma, v_j : \sigma_j \rhd M_j \rrbracket_\beta(y,\gamma',\varepsilon_j(b)),$$

we have $\llbracket \Gamma \rhd M \rrbracket_\alpha(x,\gamma)\, R^{|\sigma|}\, \llbracket \Gamma \rhd M \rrbracket_\beta(y,\gamma')$, and (1) holds for $\Gamma \rhd M : \sigma$ in this case.

The inductive arguments for terms derived by the other rules of Figure 2 follow along similar lines, and are left to the reader. ■

COROLLARY 5.3 *Suppose that R is a $|\tau|$-bisimulation from α to β, and let $\Gamma \rhd M : \varphi$ be any observable τ-formula.*

(1) *If $\gamma R^\Gamma \gamma'$ and xRy, then*

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

(2) *When $\Gamma \rhd M$ is also rigid,*

$$\text{if } xRy, \text{ then} \quad \alpha, x \models \Gamma \rhd \varphi \text{ iff } \beta, y \models \Gamma \rhd \varphi.$$

Proof.

If φ has the form $M_1 \approx M_2$, then (1) follows from part (2) of Theorem 5.2. For general φ, (1) then follows by induction on the formation of formulas. But (2) follows from (1) because $R^\Gamma = \text{id}$ when Γ is rigid. ■

By part (1) of Corollary 5.3, if an observable formula $\Gamma \rhd \varphi$ is valid in α and R is surjective, so that R^Γ is also surjective, $\Gamma \rhd \varphi$ will be valid in β. On the other hand if $\beta \models \Gamma \rhd \varphi$ and R is total, so that R^Γ is also total, then $\alpha \models \Gamma \rhd \varphi$. 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 $\Gamma \rhd \varphi$ 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 $\alpha_j \models \Gamma \rhd \varphi$ we get $\sum_I \alpha_i, \iota_j(a), \gamma \models \Gamma \rhd \varphi$, because $\alpha_j, a, \gamma \models \Gamma \rhd \varphi$, $\gamma R^\Gamma \gamma$, and the insertion morphism ι_j is a bisimulation. To sum up:

THEOREM 5.4 *The class $\{\alpha : \alpha \models \Gamma \rhd \varphi\}$ 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 $\Gamma \rhd \varphi$ is rigid and observable, then its class of models is also closed under disjoint unions.* ■

The main objective of this article 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 7.2). The key to this is the relation $\equiv_{\alpha\beta}$ defined by

$$x \equiv_{\alpha\beta} y \text{ iff every } \textit{ground observable} \text{ term } M \text{ has } [\![M]\!]_\alpha(x) = [\![M]\!]_\beta(y).$$

$\equiv_{\alpha\beta}$ is a bisimulation from α to β (Lemma 7.1), and turns out to be the largest one (Theorem 7.2). The proof of this requires the development of another characterisation of bisimulation, using the notion of "paths" between functors introduced in [18, Section 6].

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

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

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

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

- $\langle\rangle_A : TA \circ\!\!\longrightarrow TA$ is the identity function id_{TA}, so is totally defined.
- $(\pi_j.p)_A = p_A \circ \pi_j$, the composition of $T_1A \times T_2A \xrightarrow{\pi_j} T_jA \overset{p_A}{\circ\!\!\longrightarrow} SA$. Thus $x \in \mathrm{Dom}\,(\pi_j.p)_A$ iff $\pi_j(x) \in \mathrm{Dom}\, p_A$.
- $(\varepsilon_j.p)_A = p_A \circ \varepsilon_j$, the composition of $T_1A + T_2A \overset{\varepsilon_j}{\circ\!\!\longrightarrow} T_jA \overset{p_A}{\circ\!\!\longrightarrow} SA$. Thus $x \in \mathrm{Dom}\,(\varepsilon_j.p)_A$ iff $x \in \mathrm{Dom}\,\varepsilon_j$ and $\varepsilon_j(x) \in \mathrm{Dom}\, p_A$.
- $(ev_d.p)_A = p_A \circ ev_d$, the composition of $(TA)^D \xrightarrow{ev_d} TA \overset{p_A}{\circ\!\!\longrightarrow} SA$. Thus $f \in \mathrm{Dom}\,(ev_d.p)_A$ iff $f(d) \in \mathrm{Dom}\, p_A$.

A path $T \rightsquigarrow S$ is a *state path* if $S = \mathrm{Id}$, an *observation path* if $S = \bar{D}$ for some set D, and a *basic* path if it is either.

LEMMA 5.5 *Suppose T is a polynomial functor, A a set, and $x \in TA$. Then there exists a basic path $T \overset{p}{\rightsquigarrow} S$ with $x \in \mathrm{Dom}\, p_A$.*

Proof. By induction of the construction of T. If $T = \mathrm{Id}$ or $T = \bar{D}$, let p be the empty path $T \rightsquigarrow T$, so that $\mathrm{Dom}\, p_A = TA$. If $T = T_1 + T_2$, then $x \in \iota_j T_j A$ for some $j = 1, 2$. Then $\varepsilon_j(x) \in T_j A$ so by induction hypothesis there is a basic $T \overset{p}{\rightsquigarrow} S$ with $\varepsilon_j(x) \in \mathrm{Dom}\, p_A$. Then $T_1 + T_2 \overset{\varepsilon_j . p}{\rightsquigarrow} S$ is basic, with $x \in \mathrm{Dom}\, (\varepsilon_j . p)_A$.

The cases of $= T_1 \times T_2$ and $T = T_1^D$ are similarly straightforward. ■

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 5.6 *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) *$x R^T y$.*

(2) *For all paths $T \overset{p}{\rightsquigarrow} S$, $p_A(x) R^S p_B(y)$.*

(3) *For all state paths $T \overset{p}{\rightsquigarrow} \mathrm{Id}$, $p_A(x)\, R\, p_B(y)$; and for all observation paths $T \overset{p}{\rightsquigarrow} \bar{D}$, $p_A(x) = p_B(y)$.*

Proof. (1) implies (2): this is proven by induction on the construction of T. If $T = \mathrm{Id}$ or $T = \bar{D}$, then the only path p from T is the empty path $T \rightsquigarrow T$, with $p_A = \mathrm{id}_{TA}$ and $p_B = \mathrm{id}_{TB}$. Thus if (1) holds, then $p_A(x) = x\, R^T\, y = p_B(y)$, giving (2). If $T = T_1 + T_2$, then from $x R^T y$ we conclude that for some $j = 1, 2$, $x \in \iota_j T_j A$, $y \in \iota_j T_j B$, and $\varepsilon_j(x) R^{T_j} \varepsilon_j(y)$. But now a path $T \overset{p}{\rightsquigarrow} S$ must have the form $T_1 + T_2 \overset{\varepsilon_{j'} . q}{\rightsquigarrow} S$ with $T_{j'} \overset{q}{\rightsquigarrow} S$ for some $j' = 1, 2$. If $j' \neq j$, then neither $\varepsilon_j(x) \in \mathrm{Dom}\, q_A$ nor $\varepsilon_j(y) \in \mathrm{Dom}\, q_B$, so neither $p_A(x)$ nor $p_B(y)$ is defined. If $j' = j$, then by induction hypothesis on T_j, since $\varepsilon_j(x) R^{T_j} \varepsilon_j(y)$, we have $q_A(\varepsilon_j(x))$ defined iff $q_B(\varepsilon_j(y))$ is defined, and when both are defined $p_A(x) = q_A(\varepsilon_j(x)) R^S q_B(\varepsilon_j(y)) = p_B(y)$ as desired. The cases of $= T_1 \times T_2$ and $T = T_1^D$ are similarly straightforward.

That (2) implies (3) is immediate from the definitions, so it remains to show (3) implies (1). If $T = \mathrm{Id}$ or $T = \bar{D}$, let p be the empty path $T \rightsquigarrow T$. Then if (3) holds we have $x = p_A(x)\, R^T\, p_B(y) = y$, giving (1).

If $T = T_1 + T_2$, then for some j, $x \in \iota_j T_j A$ and so $\varepsilon_j(x) \in T_j A$. Now by Lemma 5.5 there exists a basic path $T_j \overset{q}{\rightsquigarrow} S$ with $\varepsilon_j(x) \in \operatorname{Dom} q_A$, so $(\varepsilon_j.q)_A = q_A(\varepsilon_j(x))$ is defined. If (3) holds for T, then applying it to the path $p = \varepsilon_j.q$ we conclude that $(\varepsilon_j.q)_B(y)$ is defined, so $y \in \iota_j T_j B$ and $\varepsilon_j(y) \in \operatorname{Dom} q_B$. But if $T_j \overset{q'}{\rightsquigarrow} S'$ is *any* basic path out of T_j, the reasoning just given shows that $q'_A(\varepsilon_j(x))$ is defined iff $q'_B(\varepsilon_j(y))$ is defined, and when they are both defined, (3) for T ensures that $q'_A(\varepsilon_j(x)) R^S q'_B(\varepsilon_j(y))$. This proves that (3) holds for T_j, so by induction hypothesis on T_j, $\varepsilon_j(x) R^{T_j} \varepsilon_j(y)$, which implies $x R^T y$ as desired. Again the cases of $T = T_1 \times T_2$ and $T = T_1^D$ are left to the reader. ■

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

THEOREM 5.7 *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*

(1) *for all state paths* $T \overset{p}{\rightsquigarrow} \mathrm{Id}$, $p_A(\alpha(x))\, R\, p_B(\beta(y))$*; and*

(2) *for all observation paths* $T \overset{p}{\rightsquigarrow} \bar{D}$, $p_A(\alpha(x)) = p_B(\beta(y))$.

Proof. By Theorem 5.6, (1) and (2) together are equivalent to $\alpha(x) R^T \beta(y)$. ■

COROLLARY 5.8 *If* $C \subseteq \operatorname{Dom}\alpha$*, then* C *is a subcoalgebra of* α *iff* $x \in C$ *implies* $p_A(\alpha(x)) \in C$ *for all state paths* $T \overset{p}{\rightsquigarrow} \mathrm{Id}$ *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 [32, 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 $\operatorname{Dom} R$ is a subcoalgebra of α. For if $x \in \operatorname{Dom} R$ and $p_A(\alpha(x))\downarrow$, then xRy for some y, so $p_A(\alpha(x))\, R\, p_B(\beta(y))$ by 5.6 and hence $p_A(\alpha(x)) \in \operatorname{Dom} R$. Similarly, the *image* of R is seen to be a subcoalgebra of β.

Theorem 5.7 also yields a characterisation of morphisms between polynomial coalgebras:

COROLLARY 5.9 *A function $f : A \to B$ is a T-morphism from (A, α) to (B, β) if, and only if,*

(1) $f(p_A(\alpha(x))) = p_B(\beta(f(x)))$ *for all state paths* $T \overset{p}{\rightsquigarrow} \mathrm{Id}$; *and*

(2) $p_A(\alpha(x)) = p_B(\beta(f(x)))$ *for all observation paths* $T \overset{p}{\rightsquigarrow} \bar{D}$.

Proof. Let R be the graph of f, i.e. xRy iff $y = f(x)$. Then f is a T-morphism iff R is a T-bisimulation. Now apply 5.7. ■

6 Definability of Path Action

We have seen that path functions are 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, a fact that may be an intuitively expected feature of our formalism, but one which requires a delicate analysis to establish.

THEOREM 6.1 *For any path $|\tau| \overset{p}{\rightsquigarrow} |\sigma|$ and variable v there exists a* tr*-free τ-term of the form*

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

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

$$p_A(\alpha(x)) = [\![\bar{p}[\mathsf{tr}(\mathsf{s})/v]]\!]_\alpha(x).$$

Proof. Note that since $\emptyset \rhd \mathsf{tr}(\mathsf{s}) : \tau$ is a τ-term, so too is $\emptyset \rhd \bar{p}[\mathsf{tr}(\mathsf{s})/v] : \sigma$ by the substitution rule (Subst). Hence $\bar{p}[\mathsf{tr}(\mathsf{s})/v]$ is a ground term of type σ.

Since the symbol tr does not occur in the raw term $\bar{p}$, the denotation $[\![v : \tau \rhd \bar{p} : \sigma]\!]_\alpha$ depends only on A, not on α, and may be written $[\![v : \tau \rhd \bar{p} : \sigma]\!]_A$.

The Theorem is proven by induction on the length of paths ending at $|\sigma|$. First, if $\tau = \sigma$ and p is the empty path $|\sigma| \rightsquigarrow |\sigma|$, let $\bar{p} = v$. It follows that $[\![\bar{p}[\mathsf{tr}(\mathsf{s})/v]]\!]_\alpha(x) = [\![\mathsf{tr}(\mathsf{s})]\!]_\alpha(x) = \alpha(x) = p_A(\alpha(x))$ since p_A is the identity function.

Now take the case $\tau = \tau_1 + \tau_2$. Then $p = \varepsilon_j.q$ for some j and some path $|\tau_j| \overset{q}{\rightsquigarrow} |\sigma|$. From the base term $v : \tau \rhd v : \tau$ we obtain a term $v : \tau \rhd \varepsilon_j v : \tau_j$ by Lemma 4.1, and by induction hypothesis there is a term $v : \tau_j \rhd \bar{q} : \sigma$ fulfilling the Theorem for q. By (Subst) these terms yield the term $v : \tau \rhd \bar{p} : \sigma$ where $\bar{p} = \bar{q}[\varepsilon_j v/v]$. Then $\bar{p}[\mathsf{tr}(\mathsf{s})/v] = \bar{q}[(\varepsilon_j v[\mathsf{tr}(\mathsf{s})/v])/v] = \bar{q}[\varepsilon_j \mathsf{tr}(\mathsf{s})/v]$ by equation (4.4). If (A, α) is a τ-coalgebra and $\alpha(x) \in \mathrm{Dom}\, p_A$, then $\alpha(x) \in \iota_j [\![\tau_j]\!]_A$ and $\varepsilon_j(\alpha(x)) \in \mathrm{Dom}\, q_A$. Let $\alpha_j : A \to [\![\tau_j]\!]_A$ be any extension of the partial function $\varepsilon_j \circ \alpha$. Then

$$
\begin{array}{ll}
[\![\bar{p}[\mathsf{tr}(\mathsf{s})/v]]\!]_\alpha(x) & \\
= [\![\bar{q}[\varepsilon_j\mathsf{tr}(\mathsf{s})/v]]\!]_\alpha(x) & \\
= [\![v:\tau_j \rhd \bar{q}:\sigma]\!]_\alpha(x, [\![\varepsilon_j\mathsf{tr}(\mathsf{s}):\tau_j]\!]_\alpha(x)) & \text{semantics of (Subst)} \\
= [\![v:\tau_j \rhd \bar{q}:\sigma]\!]_A(x, [\![\mathsf{tr}(\mathsf{s}):\tau_j]\!]_{\alpha_j}(x)) & \bar{q} \text{ is } \mathsf{tr}\text{-free, and (4.3)} \\
= [\![\bar{q}[\mathsf{tr}(\mathsf{s})/v]:\tau_j]\!]_{\alpha_j}(x) & \text{semantics of (Subst)} \\
= q_A(\alpha_j(x)) & \text{hypothesis on } q \\
= q_A(\varepsilon_j(\alpha(x))) = p_A(\alpha(x)). &
\end{array}
$$

This establishes the Theorem for the case $\tau = \tau_1 + \tau_2$.

Next suppose $\tau = \tau_1 \times \tau_2$ and so $p = \pi_j.q$ for some j and some path $|\tau_j|\overset{q}{\rightsquigarrow}|\sigma|$. Then $v:\tau \rhd \pi_j v:\tau_j$ (Proj$_j$) and $v:\tau_j \rhd \bar{q}:\sigma$ yield $v:\tau \rhd \bar{p}:\sigma$ with $\bar{p} = \bar{q}[\pi_j v/v]$ by (Subst). Moreover $\bar{p}[\mathsf{tr}(\mathsf{s})/v] = \bar{q}[\pi_j\mathsf{tr}(\mathsf{s})/v]$. This time if $\alpha(x) \in \mathrm{Dom}\, p_A$, then $\pi_j(\alpha(x)) \in \mathrm{Dom}\, q_A$. The proof of the Theorem for p now proceeds just as in the $\tau_1 + \tau_2$ case, but using the coalgebra $\pi_j \circ \alpha : A \to [\![\tau_j]\!]_A$ in place of α_j and (4.1) in place of (4.3), as well as the induction hypothesis on q.

Finally there is the case $\tau = (o \Rightarrow \tau_1)$ and $p = ev_d.q$ with $d \in [\![o]\!]$ and $|\tau_1|\overset{q}{\rightsquigarrow}|\sigma|$. From (Con) and (Weak) we get $v:\tau \rhd d:o$, and so by (App), $v:\tau \rhd v\cdot d:\tau_1$. Substitution and the hypothesis on q then allow us to put $\bar{p} = \bar{q}[v\cdot d/v]$. The rest of the proof is as in the $\tau_1 + \tau_2$ case, but using the coalgebra $ev_d \circ \alpha : A \to [\![\tau_1]\!]_A$ in place of α_j and (4.2) in place of (4.3). ■

The term function $[\![\bar{p}[\mathsf{tr}(\mathsf{s})/v]]\!]_\alpha$ from Theorem 6.1 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 observable terms that "discriminate" between the two summands of a coproduct $[\![\tau_1]\!]_A + [\![\tau_2]\!]_A$ and determine whether $p_A(\alpha(x))$ is defined. For this to work we need the (plausible) assumption that we have available at least one observable type μ that is *non-trivial* in the sense that $[\![\mu]\!]$ has at least two distinct members, say c_1 and c_2. These can be used to form the term $v:\tau_1+\tau_2 \rhd P:\mu$, where

$$P := \mathsf{case}\ v\ \mathsf{of}\ [\iota_1 v_1 \mapsto c_1 \mid \iota_2 v_2 \mapsto c_2]. \tag{6.1}$$

Then in any $\tau_1 + \tau_2$-coalgebra α, the ground term $P[\mathsf{tr}(\mathsf{s})/v]:\mu$ is a discriminator:

$$[\![P[\mathsf{tr}(\mathsf{s})/v]]\!]_\alpha(x) = c_j \quad \text{iff} \quad \alpha(x) \in \iota_j[\![\tau_j]\!]_A = \mathrm{Dom}\,\varepsilon_j. \tag{6.2}$$

This observation will be applied to establish (see Corollary 6.3) that if states x in coalgebra α and y in coalgebra β assign the same values to all ground observable terms, then $p_A(\alpha(x))$ is defined iff $p_B(\beta(y))$ is defined. The key here is the following technical result, which uses similar ideas to those found in the proof of Theorem 6.1.

LEMMA 6.2 *Suppose that τ has at least one non-trivial observable subtype μ. For any path $|\tau|\overset{p}{\rightsquigarrow}|\sigma|$ there is a finite set X_p of* tr*-free τ-terms of the form $v:\tau\rhd M:\mu$, and an associated set $X_p[\mathsf{tr}] = \{M[\mathsf{tr}(\mathsf{s})/v] : (v:\tau\rhd M)\in X_p\}$ of ground observable terms of type μ, such that for any τ-coalgebras (A,α) and (B,β) and any $x\in A$ and $y\in B$, if $[\![N]\!]_\alpha(x) = [\![N]\!]_\beta(y)$ for all $N\in X_p[\mathsf{tr}]$, then $\alpha(x)\in \mathrm{Dom}\,p_A$ iff $\beta(y)\in\mathrm{Dom}\,p_B$.*

Proof. By induction on the length of p. If $\tau=\sigma$ and p is the empty path $|\sigma|\rightsquigarrow|\sigma|$, put $X_p=\emptyset$. Then $X_p[\mathsf{tr}]=\emptyset$ and the Lemma holds (vacuously) since p_A and p_B are total (identity) functions, so $\alpha(x)\in\mathrm{Dom}\,p_A$ and $\beta(y)\in\mathrm{Dom}\,p_B$ are both true.

For the inductive case of coproducts, suppose $\tau=\tau_1+\tau_2$ and $p=\varepsilon_j.q$ for some j and some path $|\tau_j|\overset{q}{\rightsquigarrow}|\sigma|$. By induction hypothesis there is a finite set X_q of terms of the form $v:\tau_j\rhd M:\mu$ that fulfils the Lemma for q. Apply substitution (Subst) to the term $v:\tau\rhd\varepsilon_j v:\tau_j$ to generate the finite set

$$X_p = \{(v:\tau\rhd M[\varepsilon_j v/v]) \,:\, (v:\tau_j\rhd M)\in X_q\}\cup\{(v:\tau\rhd P:\mu)\},$$

with P being the raw case-term defined in line (6.1) above. Note that if $(v:\tau_j\rhd M)\in X_q$, then by equation (4.4),

$$M[\varepsilon_j\mathsf{tr}(\mathsf{s})/v] = M[(\varepsilon_j v[\mathsf{tr}(\mathsf{s})/v])/v] = (M[\varepsilon_j v/v])[\mathsf{tr}(\mathsf{s})/v]\in X_p[\mathsf{tr}].$$

Now suppose state x in τ-coalgebra α assigns the same values as y in β to all terms in $X_p[\mathsf{tr}]$. Then in particular $[\![P[\mathsf{tr}(\mathsf{s})/v]]\!]_\alpha(x) = [\![P[\mathsf{tr}(\mathsf{s})/v]]\!]_\beta(y)$, so for some $j'=1,2$, $\alpha(x)\in\iota_{j'}[\![\tau_{j'}]\!]_A$ and $\beta(y)\in\iota_{j'}[\![\tau_{j'}]\!]_B$. If $j'\neq j$, then $\alpha(x)\notin\iota_j[\![\tau_j]\!]_A$ and $\beta(y)\notin\iota_j[\![\tau_j]\!]_B$, so $\alpha(x)\notin\mathrm{Dom}\,(\varepsilon_j.q)_A$ and $\beta(y)\notin\mathrm{Dom}\,(\varepsilon_j.q)_B$, satisfying the Lemma. If however $j'=j$, let $\alpha_j: A\to[\![\tau_j]\!]_A$ be any extension of $\varepsilon_j\circ\alpha$ and $\beta_j: B\to[\![\tau_j]\!]_B$ be any extension of $\varepsilon_j\circ\beta$. For each term $v:\tau_j\rhd M$ of X_q, reasoning as in the proof of Theorem 6.1 with M in place of $\bar{q}$ shows that $[\![M[\varepsilon_j\mathsf{tr}(\mathsf{s})/v]]\!]_\alpha(x) = [\![M[\mathsf{tr}(\mathsf{s})/v]]\!]_{\alpha_j}(x)$ and likewise $[\![M[\varepsilon_j\mathsf{tr}(\mathsf{s})/v]]\!]_\beta(y) = [\![M[\mathsf{tr}(\mathsf{s})/v]]\!]_{\beta_j}(y)$. But $M[\varepsilon_j\mathsf{tr}(\mathsf{s})/v]\in X_p[\mathsf{tr}]$ and x in α agrees with y in β on $X_p[\mathsf{tr}]$, so $[\![M[\varepsilon_j\mathsf{tr}(\mathsf{s})/v]]\!]_\alpha(x) = [\![M[\varepsilon_j\mathsf{tr}(\mathsf{s})/v]]\!]_\beta(y)$. Therefore $[\![M[\mathsf{tr}(\mathsf{s})/v]]\!]_{\alpha_j}(x) = [\![M[\mathsf{tr}(\mathsf{s})/v]]\!]_{\beta_j}(y)$. This shows that x in α_j agrees with y in β_j on all terms from $X_q[\mathsf{tr}]$, so by the induction hypothesis on q, $\alpha_j(x)\in\mathrm{Dom}\,q_A$ iff $\beta_j(y)\in\mathrm{Dom}\,q_B$. But $\alpha(x)\in\iota_j[\![\tau_j]\!]_A$ and $\beta(y)\in\iota_j[\![\tau_j]\!]_B$ (as j'=j), so $\alpha_j(x)=\varepsilon_j(\alpha(x))$ and $\beta_j(y)=\varepsilon_j(\beta(y))$. It follows that $\alpha(x)\in\mathrm{Dom}\,(\varepsilon_j.q)_A$ iff $\beta(y)\in\mathrm{Dom}\,(\varepsilon_j.q)_B$, as desired, completing the proof for the case $\tau=\tau_1+\tau_2$.

Next is the case $\tau=\tau_1\times\tau_2$ with $p=\pi_j.q$ for some j and some path $|\tau_j|\overset{q}{\rightsquigarrow}|\sigma|$. With X_q given by the induction hypothesis on q, put

$X_p = \{(v : \tau \rhd M[\pi_j v/v]) : (v : \tau_j \rhd M) \in X_q\}$. Then $(v : \tau_j \rhd M) \in X_q$ implies $M[\pi_j \mathsf{tr}(\mathsf{s})/v] \in X_p[\mathsf{tr}]$, and if x in α agrees with y in β on $X_p[\mathsf{tr}]$, reasoning similar to the previous case leads to $[\![M[\mathsf{tr}(\mathsf{s})/v]]\!]_{\pi_j \circ \alpha}(x) = [\![M[\mathsf{tr}(\mathsf{s})/v]]\!]_{\pi_j \circ \beta}(y)$. Hence x in coalgebra $\pi_j \circ \alpha$ agrees with y in $\pi_j \circ \beta$ on $X_q[\mathsf{tr}]$, so by hypothesis on q, $\pi_j \circ \alpha(x) \in \mathrm{Dom}\, q_A$ iff $\pi_j \circ \beta(y) \in \mathrm{Dom}\, q_B$, i.e. $\alpha(x) \in \mathrm{Dom}\, (\pi_j.q)_A$ iff $\beta(y) \in \mathrm{Dom}\, (\pi_j.q)_B$.

Finally there is the case $\tau = (o \Rightarrow \tau_1)$ and $p = ev_d.q$ with $d \in [\![o]\!]$ and $|\tau_1| \overset{q}{\rightsquigarrow} |\sigma|$. This time $X_p = \{(v : \tau \rhd M[v \cdot d/v]) : (v : \tau_1 \rhd M) \in X_q\}$. Similarly to the $\tau_1 + \tau_2$ case, we get that if x in α agrees with y in β on $X_p[\mathsf{tr}]$, then x in coalgebra $ev_d \circ \alpha$ agrees with y in $ev_d \circ \beta$ on $X_q[\mathsf{tr}]$, so $ev_d \circ \alpha(x) \in \mathrm{Dom}\, q_A$ iff $ev_d \circ \beta(y) \in \mathrm{Dom}\, q_B$, i.e. $\alpha(x) \in \mathrm{Dom}\, (ev_d.q)_A$ iff $\beta(y) \in \mathrm{Dom}\, (ev_d.q)_B$. ∎

COROLLARY 6.3 *Let (A,α) and (B,β) be τ-coalgebras, where τ has at least one non-trivial observable subtype μ. Let $x \in A$ and $y \in B$ have $[\![N]\!]_\alpha(x) = [\![N]\!]_\beta(y)$ for all ground τ-terms N of type μ. Then for any path $|\tau| \overset{p}{\rightsquigarrow} |\sigma|$, $\alpha(x) \in \mathrm{Dom}\, p_A$ iff $\beta(y) \in \mathrm{Dom}\, p_B$.* ∎

In a similar vein to Lemma 6.2, we can define *formulas* that characterise the property of belonging to the domain of a path function:

LEMMA 6.4 *Suppose that τ has at least one non-trivial observable subtype μ. For any path $|\tau| \overset{p}{\rightsquigarrow} |\sigma|$ and variable v there exists a* tr*-free τ-formula $v : \tau \rhd \varphi_p$ such that for any τ-coalgebra (A,α) and any $x \in A$,*

$$\alpha, x \models \varphi_p[\mathsf{tr}(\mathsf{s})/v] \quad \textit{iff} \quad \alpha(x) \in \mathrm{Dom}\, p_A.$$

Proof. If p is the empty path, then $\alpha(x) \in \mathrm{Dom}\, p_A$ for all $x \in A$. In that case let φ_p be the equation $c \approx c$ for some $c \in [\![\mu]\!]$. Then $\varphi_p[\mathsf{tr}(\mathsf{s})/v]$ is also $c \approx c$, which is satisfied in α at all $x \in A$.

For the inductive case of coproducts, suppose $\tau = \tau_1 + \tau_2$ and $p = \varepsilon_j.q$ for some j and some path $|\tau_j| \overset{q}{\rightsquigarrow} |\sigma|$. Let φ_p be $(P \approx c_j) \wedge \varphi_q[\varepsilon_j v/v]$, where P is the discriminating case-term of line (6.1), and φ_q is given by the induction hypothesis as fulfilling the Lemma for q. Now $\alpha(x) \in \mathrm{Dom}\, p_A$ iff $\alpha(x) \in \iota_j[\![\tau_j]\!]_A$ and $\varepsilon_j \circ \alpha(x) \in \mathrm{Dom}\, q_A$. But by line (6.2), $\alpha(x) \in \iota_j[\![\tau_j]\!]_A$ iff $\alpha, x \models P[\mathsf{tr}(\mathsf{s})/v] \approx c_j$. Also, if $\alpha_j : A \to [\![\tau_j]\!]_A$ is any extension of $\varepsilon_j \circ \alpha$, and $\varepsilon_j \circ \alpha(x)$ is defined, then by induction hypothesis on q, $\varepsilon_j \circ \alpha(x) \in \mathrm{Dom}\, q_A$ iff $\alpha_j, x \models \varphi_q[\mathsf{tr}(\mathsf{s})/v]$, which holds iff $\alpha, x \models \varphi_q[\varepsilon_j \mathsf{tr}(\mathsf{s})/v]$ by (4.7). So $\alpha(x) \in \mathrm{Dom}\, p_A$ iff the formula

$$(P[\mathsf{tr}(\mathsf{s})/v] \approx c_j) \wedge \varphi_q[\varepsilon_j \mathsf{tr}(\mathsf{s})/v]$$

is satisfied in α at x. But this formula is $\varphi_p[\mathsf{tr}(\mathsf{s})/v]$ (see (4.4)), so the Lemma holds for the case $\tau = \tau_1 + \tau_2$.

If $\tau = \tau_1 \times \tau_2$ with $p = \pi_j.q$ for some j and some path $|\tau_j| \overset{q}{\rightsquigarrow} |\sigma|$, let φ_p be $\varphi_q[\pi_j v/v]$. Now $\alpha(x) \in \mathrm{Dom}\, p_A$ iff $\pi_j \circ \alpha(x) \in \mathrm{Dom}\, q_A$ iff $\pi_j \circ \alpha, x \models \varphi_q[\mathsf{tr}(\mathsf{s})/v]$ (by hypothesis on q) iff $\alpha, x \models \varphi_q[\pi_j \mathsf{tr}(\mathsf{s})/v]$ by (4.5). But $\varphi_q[\pi_j \mathsf{tr}(\mathsf{s})/v]$ is $\varphi_p[\mathsf{tr}(\mathsf{s})/v]$ in this case.

If $\tau = (o \Rightarrow \tau_1)$ and $p = ev_d.q$ with $d \in [\![o]\!]$ and $|\tau_1| \overset{q}{\rightsquigarrow} |\sigma|$, let φ_p be $\varphi_q[v \cdot d/v]$. Then analogously to the previous case, but using (4.6), we get $\alpha(x) \in \mathrm{Dom}\, p_A$ iff $ev_d \circ \alpha(x) \in \mathrm{Dom}\, q_A$ iff $ev_d \circ \alpha, x \models \varphi_q[\mathsf{tr}(\mathsf{s})/v]$ iff $\alpha, x \models \varphi_q[\mathsf{tr}(\mathsf{s}) \cdot d/v]$ as desired. ∎

Definable Modalities

The terms provided by Theorem 6.1 can be used to formalise modal assertions that certain formulas will be true after the execution of state transitions induced by path functions. This is a special case of the following result, which shows that we can express more general modal assertions associated with state transitions $x \mapsto [\![M]\!]_\alpha(x)$ defined by any term M of state type.

THEOREM 6.5 *If M is any ground term of type* St*, and φ any ground formula, then in any τ-coalgebra (A, α),*

$$\alpha, [\![M]\!]_\alpha(x) \models \varphi \quad \textit{iff} \quad \alpha, x \models \varphi[M/\mathsf{s}].$$

Proof. By induction on the formation of φ, with the inductive cases of the connectives $\neg$ and $\wedge$ being straightforward, so it suffices to consider the case that φ is an equation $N_1 \approx N_2$. For $i = 1, 2$, $[\![N_i]\!]_\alpha([\![M]\!]_\alpha(x))$ equals $[\![N_i[M/s]]\!]_\alpha(x)$ by the semantics of (s-Subst). Hence

$$\begin{array}{l} [\![N_1]\!]_\alpha([\![M]\!]_\alpha(x)) = [\![N_2]\!]_\alpha([\![M]\!]_\alpha(x))) \quad \text{iff} \\ [\![N_1[M/s]]\!]_\alpha(x) = [\![N_2[M/s]]\!]_\alpha(x), \end{array}$$

which proves the Theorem for this case of φ. ∎

Thus the formula $\varphi[M/\mathsf{s}]$ is seen to expresses the modal assertion "after action M, φ". The above-mentioned case of state transitions induced by path functions is given by

COROLLARY 6.6 *Let $|\tau| \overset{p}{\rightsquigarrow} \mathrm{Id}$ be a state path and φ any ground formula. Then in any τ-coalgebra (A, α), if $\alpha(x) \in \mathrm{Dom}\, p_A$,*

$$\alpha, p_A(\alpha(x)) \models \varphi \quad \textit{iff} \quad \alpha, x \models \varphi[M_p/\mathsf{s}],$$

where M_p is the ground term $\bar{p}[\mathsf{tr}(\mathsf{s})/v]$ of type St *given by Theorem 6.1.*

Proof. From 6.1 and 6.5, as $p_A(\alpha(x)) = [\![M_p]\!]_\alpha(x)$. ■

Morphisms and Term-Value Preservation

To round out this discussion of definability of path action, here is a characterisation of coalgebraic morphisms as functions preserving values of certain terms.

THEOREM 6.7 *A function $f : A \to B$ between τ-coalgebras (A, α) and (B, β) is a $|\tau|$-morphism if, and only if,*

(1) $f([\![M]\!]_\alpha(x)) = [\![M]\!]_\beta(f(x))$ *for all ground τ-terms M of type* St; *and*

(2) $[\![M]\!]_\alpha(x) = [\![M]\!]_\beta(f(x))$ *for all ground τ-terms M of observable type.*

Proof. If f is a morphism, then the relation xRy iff $y = f(x)$ is a bisimulation, so (1) and (2) follow from Theorem 5.2 parts (1) and (2).

For the converse it is enough to know that (1) and (2) hold whenever M is the ground term $\bar{p}[\mathsf{tr}(\mathsf{s})/v]$, given by Theorem 6.1, for the cases that p is a state path or an observation path, respectively. For then (1) gives $f(p_A(\alpha(x))) = p_B(\beta(f(x)))$ for any state path, while (2) gives $p_A(\alpha(x)) = p_B(\beta(f(x)))$ for any observation path. But by Corollary 5.9, this is enough to make f is a $|\tau|$-morphism. ■

7 Logical Characterisation of Bisimilarity

Recall the relation $\equiv_{\alpha\beta}$ between the state sets of two τ-coalgebras, which has $x \equiv_{\alpha\beta} y$ iff $[\![M]\!]_\alpha(x) = [\![M]\!]_\beta(y)$ for every ground term M of observable type. We are now ready to show that $\equiv_{\alpha\beta}$ is identical to the bisimilarity relation between α and β, and that it has a characterisation in terms of satisfaction of observable formulas. The first step is

LEMMA 7.1 *If τ has at least one non-trivial observable subtype, then $\equiv_{\alpha\beta}$ is a bisimulation from α to β.*

Proof. We use the characterisation of bisimulations given by Theorem 5.7.

Let $x \equiv_{\alpha\beta} y$ and suppose $|\tau| \overset{p}{\rightsquigarrow} |\sigma|$ is a basic path from $|\tau|$. Since x and y assign the same values to all ground observable terms, it is immediate from Corollary 6.3 that $p_A(\alpha(x))$ is defined iff $p_B(\beta(y))$ is defined. Suppose then that they are both defined.

(i) If p is a state path, i.e. $\sigma = \mathsf{St}$ and $|\sigma| = \mathrm{Id}$, we must show $p_A(\alpha(x)) \equiv_{\alpha\beta} p_B(\beta(y))$. Let $v : \tau \rhd \bar{p} : \sigma$ be the term given by Theorem 6.1, and M the ground term $\bar{p}[\mathsf{tr}(\mathsf{s})/v]$ of type St. Then $p_A(\alpha(x)) = [\![M]\!]_\alpha(x) \in A$

and $p_B(\beta(y)) = [\![M]\!]_\beta(y) \in B$. Now if $N : o$ is any ground observable term, then the rule (s-Subst) gives the ground term $N[M/\mathsf{s}] : o$, and so as $x \equiv_{\alpha\beta} y$ we get $[\![N[M/\mathsf{s}]]\!]_a(x) = [\![N[M/\mathsf{s}]]\!]_\beta(y)$. Hence by the semantics of (s-Subst), $[\![N]\!]_\alpha([\![M]\!]_\alpha(x)) = [\![N]\!]_\beta([\![M]\!]_\beta(y))$, i.e. $[\![N]\!]_\alpha(p_A(\alpha(x))) = [\![N]\!]_\beta(p_B(\beta(y)))$. Since this holds for all ground observable N, it means that $p_A(\alpha(x)) \equiv_{\alpha\beta} p_B(\beta(y))$ as required.

(ii) If p is an observation path, i.e. $\sigma \in \mathbb{O}$, then $\bar{p}[\mathsf{tr}(\mathsf{s})/v]$ is of observable type. Hence as $x \equiv_{\alpha\beta} y$, $[\![\bar{p}[\mathsf{tr}(\mathsf{s})/v]]\!]_a(x) = [\![\bar{p}[\mathsf{tr}(\mathsf{s})/v]]\!]_\beta(y)$, so by Theorem 6.1 $p_A(\alpha(x)) = p_B(\beta(y))$ as required.

This completes the proof that $\equiv_{\alpha\beta}$ fulfills the criterion of Theorem 5.7 for being a bisimulation. ■

Here now is our logical characterisation of bisimilarity for polynomial coalgebras:

THEOREM 7.2 *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) *$\alpha, x \models \varphi$ iff $\beta, y \models \varphi$ for all ground observable formulas φ.*

(4) *$\alpha, x \models M \approx N$ iff $\beta, y \models M \approx N$ for all ground observable terms M and N.*

(5) *$\alpha, x \models M \approx N$ implies $\beta, y \models M \approx N$ for all ground observable terms M and N.*

(6) *$[\![M]\!]_\alpha(x) = [\![M]\!]_\beta(y)$ for all ground observable terms M, i.e. $x \equiv_{\alpha\beta} y$.*

Proof. (1) implies (2): let $x \sim y$. Thus xRy for some $|\tau|$-bisimulation R from α to β. Then (2) is given by Corollary 5.3(2). It is immediate that (2) implies (3), (3) implies (4), and (4) implies (5). If (5) holds and $c = [\![M]\!]_\alpha(x)$, where M has type $o \in \mathbb{O}$, then we can take N to be c in (5) to deduce $\beta, y \models M \approx c$, so $[\![M]\!]_\beta(y) = c = [\![M]\!]_\alpha(x)$, i.e. (6) holds. Finally, since $\equiv_{\alpha\beta}$ is a bisimulation and $\sim$ is the union of all bisimulations, $x \equiv_{\alpha\beta} y$ implies $x \sim y$, i.e. (6) implies (1). ■

8 Comparison With Other Formalisms

For each functor $T : \mathbf{Set} \to \mathbf{Set}$ we can ask for a formalism of syntax and semantics that gives a logical characterisation of the T-bisimilarity relations between T-coalgebras: states are bisimilar precisely when they are logically indistinguishable in terms of satisfaction of formulas, or of valuation of terms. This programme, initiated by Hennessy and Milner [12, 13], has been carried out for many classes of coalgebras presented as labelled transition systems associated with theories of process algebra [23, 24, 25, 11], using some species of finitary or infinitary modal logic. Moss [26] showed that infinitary modal logic is also the appropriate logical formalism for characterising bisimilarity for very general functors acting on classes.

A characterisation of bisimilarity by term-valuation appears in the theory of hidden algebras in the work of Reichel and Malcolm (see [22]). Two elements of a hidden algebra are *behaviourally equivalent* if they assign the same values to all terms having a single variable of hidden sort. This relation proves to be equivalent to bisimilarity for a representation of hidden algebras as coalgebras of certain functors constructed out of products and powers [22, Example 2, Section 4.1].

Kurz [21] developed a finitary modal language for polynomial functors having the form $T(X) = \prod_{i=1}^{n}(B_i + C_i \times X)^{A_i}$. Rößiger [30] devised another kind of finitary language that is equivalent to Kurz's for the particular functors just mentioned, but is general enough to be able to characterise bisimilarity for all polynomial functors. It uses the notion of a *position*, which corresponds to what we have called a *basic path*. For each state path $T \overset{p}{\rightsquigarrow} \mathrm{Id}$, Rößiger introduces a modal connective $[p]$, allowing formation of formulas $[p]\varphi$. For each observation path $T \overset{p}{\rightsquigarrow} \bar{D}$, there is a formula $(p)c$ for each $c \in D$. Formulas in general are constructed from these atomic formulas $(p)c$ by applying the Boolean connectives and the modalities $[p]$. The semantics of [30, Definition 2.2] stipulates that

$$\begin{array}{lll} \alpha, x, \models (p)c & \text{iff} & p_A(\alpha(x)) \text{ is defined and } p_A(\alpha(x)) = c, \\ \alpha, x, \models [p]\varphi & \text{iff} & \text{either } p_A(\alpha(x)) \text{ is undefined or } \alpha, p_A(\alpha(x)) \models \varphi. \end{array}$$

Now let M_p be the term $\bar{p}[\mathsf{tr}(\mathsf{s})/v]$ of Theorem 6.1 having $p_A(\alpha(x)) = [\![M_p]\!]_\alpha(x)$, and let $p\!\downarrow$ be the formula $\varphi_p[\mathsf{tr}(\mathsf{s})/v]$ of Lemma 6.4 having

$$\alpha, x \models p\!\downarrow \quad \text{iff} \quad \alpha(x) \in \mathrm{Dom}\, p_A.$$

Then it is evident that $(p)c$ has the same semantics as $p\!\downarrow \wedge (M_p \approx c)$, while by Corollary 6.6 $[p]\varphi$ has the same semantics as $(p\!\downarrow \to \varphi[M_p/\mathsf{s}])$. In this way the language of [30] can be identified with a subset of the language of the present paper. Rößiger's language does allow bisimilarity to be characterised by formula satisfaction, but it appears that the stronger syntax of terms developed here is required to give a charactersation in terms of equality of term-evaluation.

A complete deductive calculus of observational equations for monomial coalgebras is presented in [5, 6]. Another equational approach is that of [4, 3], which has "coterms" generated from symbols for operations of the form $X \to X_1 + \cdots + X_n$. Polynomial coalgebras can be represented as coalgebras built from such operations [3, Section 5], but an advantage of the present approach is that it provides a suitable syntax for coalgebras in the form in which they are given, rather than one for an equivalent representing coalgebra.

In summary, the language developed in this paper has all the expressive power of these other approaches and in addition provides a natural syntax for polynomial coalgebraic operations (projections, pairings, injections, case-formations, evaluations, lambda abstractions, functional applications). This makes it possible to characterise bisimilar states as those assigning the same values to terms of observable type. While the language is close in style to those of universal algebra and categorical logic, it has also been shown to be a species of modal logic. The idea of a term or formula having a single state-valued parameter is an inherently modal one: the standard translation of propositional modal logic into first-order logic associates with each modal formula a first-order formula that has a single free variable ranging over worlds or states [20, p. 234]. Moreover, our language is capable of defining certain modalities: as the above discussion and 6.5 and 6.6 show, the formula $\varphi[M_p/\mathsf{s}]$ expresses the modal assertion "after execution of the state transition $p_A \circ \alpha$, φ will be true", while more generally, $\varphi[M/\mathsf{s}]$ expresses "φ will be true after execution of the transition $x \mapsto [\![M]\!]_\alpha(x)$".

9 Towards Birkhoff's Theorem

This paper confirms the fundamental role played by observable formulas in the theory of polynomial coalgebras, a role that could be said to be analogous to that played by equations in the theory of universal algebras. It is natural then to seek a characterisation of those classes of coalgebras that are defined by observational formulas. An approach to this has been developed in [10], and involves a new kind of ultrapower construction on coalgebras, which we now briefly describe.

Let U be an ultrafilter on a set I, i.e. U is a collection of subsets of I that is closed under finite intersections and supersets, and contains, for each $J \subseteq I$, exactly one of J and its complement $I - J$. Then for any 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 $f^U = \{g \in A^I : f =_U g\}$, and the quotient set $A^U = \{f^U : f \in A^I\}$ is called the *ultrapower of A modulo U*. Any function $\theta : A \to B$ has a *U-lifting* $\theta^U : A^U \to B^U$ given by $\theta^U(f^U) = (\theta \circ f)^U$.

Thus a τ-coalgebra $\alpha : A \to |\tau|A$ has the U-lifting $\alpha^U : A^U \to (|\tau|A)^U$, which is *not* a τ-coalgebra on A^U, since the latter would be a function of the form $A^U \to |\tau|(A^U)$. However a natural coalgebraic structure associated with A^U can be obtained by removing some of its points. Define f^U to be *observable* if for every ground observable τ-term $M : o$ there is an element $c \in [\![o]\!]$ such that the set $\{i \in I : [\![M]\!]_\alpha(f(i)) = c\}$ belongs to U. In essence this means that the U-lifting of the denotation $[\![M]\!]_\alpha : A \to [\![o]\!]$ gives f^U the value c.

If A^+ is the set of observable elements of A^U, then a coalgebra

$$\alpha^+ : A^+ \to [\![\tau]\!]_{A^+}$$

can be constructed such that if $\Gamma \rhd \varphi$ is any observable formula, then

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

We call α^+ the *observational ultrapower* of α modulo U. Its definition, and the proof that it preserves validity, involve a lengthy and complex analysis of the U-liftings of the partial functions p_A induced by paths $|\tau| \overset{p}{\rightsquigarrow} |\sigma|$.

By choosing the right kind of ultrafilter U it can be arranged that α^+ is sufficiently "saturated" with states that the following holds:

> if every ground observable τ-formula valid in α is valid also in a τ-coalgebra β, then the bisimilarity relation from α^+ to β is surjective.

The proof makes use of our Theorem 7.2. The result itself is then used to give a structural characterisation of logically definable classes of coalgebras:

(†) *a class K of coalgebras is the class of all models of some set of rigid observable formulas if, and only if, K is closed under disjoint unions, images of bisimulations and observational ultrapowers.*

This may be viewed as an analogue for polynomial coalgebras of Birkhoff's famous characterisation of equational classes of abstract algebras as being those closed under direct products, homomorphic images and subalgebras. Details of the proof, and of the construction of α^+, are given in the paper [10].

Instead of working with ultrapowers, a notion of *ultrafilter enlargement* can be developed for a coalgebras, constructing from α a new coalgebra $E\alpha$ whose states are certain ultrafilters on the state set A of α. In modal model theory a construction like this is used that is based on the set of *all* ultrafilters on A, but for polynomial coalgebras we have to restrict to certain ultrafilters F that are *observationally rich.* This means that

> for each ground observable term $M : o$ there exists some element $c \in [\![o]\!]$ such that $\{x \in A : [\![M]\!]_\alpha(x) = c\} \in F$.

If EA is the set of rich ultrafilters on A, then for any observational ultrapower (A^+, α^+) a map $\Phi : A^+ \to EA$ is given by

$$\Phi(f^U) = \{X \subseteq A : \{i : f(i) \in X\} \in U\}.$$

A transition structure $E\alpha : EA \to [\![\tau]\!]_{EA}$ can be defined such that Φ becomes a τ-morphism from α^+ to $E\alpha$. For saturated α^+, Φ is surjective. A fundamental feature of this construction is that for any $F \in EA$ and any ground observable formula φ,

$$E\alpha, F \models \varphi \quad \text{iff} \quad \varphi^\alpha \in F,$$

where $\varphi^\alpha = \{x \in A : \alpha, x \models \varphi\}$ is the subset of A *defined by* φ. This may be interpreted as saying that φ is true in $E\alpha$ at state F iff it is true in α at a set of states that is "large in the sense of F". The result is proved by transferring properties of α^+ to $E\alpha$ by the map Φ.

Yet another approach is to restrict attention to definable subsets of (A, α). The sets φ^α form a Boolean algebra, and we define ΔA to be the set of all observationally rich ultrafilters of this algebra. There is a natural map $\theta_\alpha : EA \to \Delta A$ that restricts each rich ultrafilter F on A to its members of the form φ^α. A transition $\Delta\alpha : \Delta A \to [\![\tau]\!]_{\Delta A}$ can be defined such that θ_α becomes a τ-morphism from $E\alpha$ to $\Delta\alpha$.

These constructions are related by a commuting diagram

$$\begin{array}{ccccc}
A^+ & \xrightarrow{\ \Phi\ } & EA & \xrightarrow{\ \theta_\alpha\ } & \Delta A \\
\Big\downarrow{\scriptstyle \alpha^+} & & \Big\downarrow{\scriptstyle E\alpha} & & \Big\downarrow{\scriptstyle \Delta\alpha} \\
[\![\tau]\!]_{A^+} & \xrightarrow{\ |\tau|\Phi\ } & [\![\tau]\!]_{EA} & \xrightarrow{\ |\tau|\theta_\alpha\ } & [\![\tau]\!]_{\Delta A}
\end{array}$$

$(EA, E\alpha)$ is the *ultrafilter enlargement* of α, and $(\Delta A, \Delta\alpha)$ is the *definable enlargement.* It turns out that θ_α identifies two points of EA iff they are bisimilar, so in fact $\Delta\alpha$ is essentially the quotient of $E\alpha$ by bisimilarity. A version of the definable enlargement for monomial coalgebras was discussed in [8].

Now the above analogue (†) of Birkhoff's theorem remains true if "closed under observational ultrapowers" is replaced by "closed under ultrafilter enlargements", or "closed under definable enlargements". Details of the proof, and of the construction of $\Delta\alpha$, are given in the paper [7].

BIBLIOGRAPHY

[1] Peter Aczel and Nax Mendler. A final coalgebra theorem. In D. H. Pitt et al., editors, *Category Theory and Computer Science. Proceedings 1989*, volume 389 of *Lecture Notes in Computer Science*, pages 357–365. Springer-Verlag, 1989.

[2] Garret Birkhoff. On the structure of abstract algebras. *Proceedings of the Cambridge Philosophical Society*, 31:433–454, 1935.

[3] Corina Cîrstea. A coalgebraic equational approach to specifying observational structures. *Theoretical Computer Science*, to appear.

[4] Corina Cîrstea. A coequational approach to specifying behaviours. *Electronic Notes in Theoretical Computer Science*, 19, 1999. `http://www.elsevier.nl/locate/entcs`.

[5] Andrea Corradini. A complete calculus for equational deduction in coalgebraic specification. Technical Report SEN-R9723, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, 1997.

[6] Andrea Corradini. A completeness result for equational deduction in coalgebraic specification. In Francesco Parisi Presicce, editor, *Recent Trends in Algebraic Development Techniques*, volume 1376 of *Lecture Notes in Computer Science*, pages 190–205. Springer-Verlag, 1998.

[7] Robert Goldblatt. Enlargements of polynomial coalgebras. Manuscript available at `http://www.mcs.vuw.ac.nz/~rob`.

[8] Robert Goldblatt. Duality for some categories of coalgebras. *Algebra Universalis*, 46(3):389–416, 2001.

[9] Robert Goldblatt. What is the coalgebraic analogue of Birkhoff's variety theorem? *Theoretical Computer Science*, 266:853–886, 2001.

[10] Robert Goldblatt. Observational ultraproducts of polynomial coalgebras. Manuscript available at `http://www.mcs.vuw.ac.nz/~rob`

[11] M. Hennessy and X. Liu. A modal logic for message passing processes. *Acta Informatica*, 32:375–393, 1995.

[12] Matthew Hennessy and Robin Milner. On observing nondeterminism and concurrency. In J. W. de Bakker and J. van Leeuwen, editors, *Automata, Languages and Programming. Proceedings 1980*, volume 85 of *Lecture Notes in Computer Science*, pages 299–309. Springer-Verlag, 1980.

[13] Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the Association for Computing Machinery*, 32:137–161, 1985.

[14] Claudio Hermida. *Fibrations, Logical Predicates and Indeterminates.* PhD thesis, University of Edinburgh, 1993. Techn. rep. LFCS-93-277. Also available as Aarhus Univ. DAIMI Techn. rep. PB-462.

[15] Claudio Hermida and Bart Jacobs. Structural induction and coinduction in a fibrational setting. *Information and Computation*, 145:107–152, 1998.

[16] Bart Jacobs. Objects and classes, coalgebraically. In B. Freitag, C. B. Jones, C. Lengauer, and H.-J. Schek, editors, *Object-Orientation with Parallelism and Persistence*, pages 83–103. Kluwer Academic Publishers, 1996.

[17] Bart Jacobs. *Categorical Logic and Type Theory.* Elsevier, 1999.

[18] Bart Jacobs. Towards a duality result in coalgebraic modal logic. *Electronic Notes in Theoretical Computer Science*, 33, 2000. `http://www.elsevier.nl/locate/entcs`.

[19] Bart Jacobs. Exercises in coalgebraic specification. In R. Backhouse, R. Crole, and J. Gibbons, editors, *Algebraic and Coalgebraic Methods in the Mathematics of Program Construction*, volume 2297 of *Lecture Notes in Computer Science*, pages 237–280. Springer, 2002.

[20] Marcus Kracht. *Tools and Techniques in Modal Logic*, volume 142 of *Studies in Logic*. Elsevier, 1999.

[21] Alexander Kurz. Specifying coalgebras with modal logic. *Theoretical Computer Science*, 260:119–138, 2001.

[22] Grant Malcolm. Behavioural equivalence, bisimulation, and minimal realisation. In Magne Haveraaen, Olaf Owe, and Ole-Johan Dahl, editors, *Recent Trends in Data Type Specification*, volume 1130 of *Lecture Notes in Computer Science*, pages 359–378. Springer-Verlag, 1996.

[23] Robin Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25:267–310, 1983.

[24] Robin Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[25] Robin Milner, Joachim Parrow, and David Walker. Modal logics for mobile processes. *Theoretical Computer Science*, 114:149–171, 1993.

[26] Lawrence S. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96:277–317, 1999.

[27] David Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Theoretical Computer Science*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, 1981.

[28] Andrew M. Pitts. 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*, chapter 2. Oxford University Press, 2000.

[29] Horst Reichel. An approach to object semantics based on terminal co-algebras. *Mathematical Structures in Computer Science*, 5:129–152, 1995.

[30] Martin Rößiger. From modal logic to terminal coalgebras. *Theoretical Computer Science*, 260:209–228, 2001.

[31] J.J.M.M. Rutten. A calculus of transition systems (towards universal coalgebra). In Alban Ponse, Maarten de Rijke, and Yde Venema, editors, *Modal Logic and Process Algebra*, CSLI Lecture Notes No. 53, pages 231–256. CSLI Publications, Stanford, California, 1995.

[32] J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000.