Observational Truth as a Categorical Modality

Grant Malcolm

Department of Computer Science
University of Liverpool
PO Box 147
Liverpool L69 3BX
UK

email grant@csc.liv.ac.uk

Abstract

This paper examines the notion of truth-up-to-observability in the setting of categorical logic and shows that it can be captured by a modal operator. Our main results extend the Kripke-Beth-Joyal semantics for the logics of presheaf toposes to the observational modality. We also give a categorical account of coinduction (or bisimulation) as a proof-technique for establishing observational truth.

We assume familiarity with basic notions from category theory, including: functor, natural transformation, subcategory and subfunctor.

Keywords: modal logics, categorical logic, topos theory, Kripke-Beth-Joyal semantics, behavioural equality, observational equality, hidden algebra.

Observational Truth as a Modality

1 Introduction

The notions of observation and behaviour have played an important rôle in computer science, from Nerode equivalence and minimal realisations of automata, to bisimulations in process algebras [21] and in coalgebras, particularly with respect to the object paradigm [1, 13, 25, 20], while notions of observational or behavioural equivalence have formed a continuous thread through the area of algebraic specification [24, 26, 23, 3, 10]. In this paper we present a categorical logic in which the notion of behavioural satisfaction (or *truth-up-to-observability*) can be abstracted as a modality.

The logic in question is the internal logical language of a topos of presheaves; modal operators in such languages correspond to topologies and can be read as stating that properties are 'locally true' (following Lawvere [18]). The semantics of these modalities have been developed by a number of authors, most notably for our purposes Goldblatt [11] and Ghilardi and Meloni [5]. Our own interest in these languages arose from a result of Worrell's stating that categories of models of *hidden algebraic* specifications form a topos [27] (the result holds more generally for coalgebras of certain functors [15]). We give a brief overview of hidden algebra Section 2 below. Since hidden algebra allows hierarchically structured specifications, it is of interest to examine to what extent the logical languages of the toposes of models can also be hierarchically structured.

This, however, is beyond the scope of the present paper, which focusses on how behavioural equivalence is manifested in such toposes. In Worrell's toposes, behavioural equivalence is 'built-in', in the sense that limits in those toposes are constructed from behaviourally equivalent elements (e.g., products *are* the relation of behavioural equivalence). We investigate a similar topos structure in Section 3 and define a modality that captures behavioural equivalence. We also give a categorical account of coinduction, a proof technique commonly used in hidden algebra and coalgebra [9, 22, 14].

2 Hidden Algebra

In order to motivate the constructions of Section 3, we give a brief introduction to behavioural equality in the context of hidden algebra, which was introduced by Goguen [6] as a foundation for a semantics of the object paradigm. Hidden algebra is based on many-sorted equational logic, but has a weaker notion of satisfaction of equations: equations are behaviourally satisfied if no experiment can distinguish between the left- and right-hand sides of an equation. As a specification formalism, hidden algebra therefore allows a wider class of correct implementations (i.e., models), wich include all the implementations that *seem* to satisfy the equations in a specification.

Here we only sketch the details of hidden algebra; a full account can be found in [10]. We begin with the following

Example 1 As an example, consider the following specification of a Flag object, which has one boolean attribute that says whether the flag is up or down, and has methods to raise and lower the flag, and to reverse it (put it up if it's down and vice-versa):

```
bth FLAG is
  hsort Flag .
  ops  up dn rev : Flag -> Flag .
  op   up? : Flag -> Bool .
  var F : Flag .
  eq  up?(up(F))  =  true .
  eq  up?(dn(F))  =  false .
  eq  up?(rev(F)) =  not(up?(F)) .
end
```

This specification uses the notation of BOBJ [8]. The key feature is that there are *hidden* and *visible* sorts; in this case, `Flag` is hidden and `Bool` is visible. In general, visible sorts have a fixed interpretation (e.g., the Booleans) and hidden sorts are intended to represent the states of objects (e.g., a Flag object). □

A model (i.e., a **hidden algebra**) of such a specification is an algebra (i.e., sorts are interpreted as sets; operations as appropriately typed functions) that has a fixed interpretation of the visible sorts (e.g., if the interpretation of `Bool` is fixed as $\{0, 1\}$, then any hidden algebra interprets `Bool` as that set.

The notion of behavioural satisfaction of equations uses the notion of **context**, which is a term with exactly one occurrence of a placeholder variable (say, '`_`'); visible contexts are contexts of some visible sort. For example, the visible contexts for `Flag` include:

```
up?(_), up?(up(_)), up?(dn(_)), up?(rev(_)), up?(up(dn(_))).
```

We write $\mathsf{C}_\Sigma(s,t)$ for the set of contexts of sort t that contain a placeholder variable of sort s; if $c \in \mathsf{C}_\Sigma(s,t)$ and e is a term of sort s, we write $c[e]$ for the term obtained by substituting e for the placeholder variable in c.

A hidden algebra **behaviourally satisfies** an equation $(\forall X)\ l = r$ iff for all visible contexts c, it satisfies the equation $(\forall X)\ c[l] = c[r]$. For example, a hidden `FLAG`-algebra behaviourally satisfies the equation

$(\forall F$ `: Flag) rev(rev(`F`)) =` F

iff it satisfies all the equations

$(\forall F$ `: Flag) up?(rev(rev(`F`))) = up?(`F`)`
$(\forall F$ `: Flag) up?(up(rev(rev(`F`)))) = up?(up(`F`))`
$(\forall F$ `: Flag) up?(dn(rev(rev(`F`)))) = up?(dn(`F`))`

i.e., an infinite number of equations need to be satisfied, which can make proving behavioural satisfaction tedious (e.g., by induction on the structure of contexts [12]). The principle of **coinduction** is a powerful tool in proving behavioural satisfaction; its statement uses the notion of behavioural equivalence for elements of a hidden algebra's carrier set. Two elements x and y are behaviourally equivalent iff $c[x] = c[y]$ for all visible contexts c. This relation is clearly a congruence relation, and for x and y of visible sort, it is clearly the relation of equality; we call any such congruence relation whose restriction to visible sorts is equality a **behavioural congruence**. In fact, we have the following (see [10]):

Theorem 2 For any hidden algebra A, behavioural equivalence is the largest behavioural congruence on A. $\square$

Clearly, a hidden algebra behaviourally satisfies an equation iff the left- and right-hand sides are behaviourally equivalent, so the principle of coinduction says that to prove behavioural satisfaction, it is sufficient to find some behavioural congruence that relates the left- and right-hand sides.

Example 3 To prove that a hidden `FLAG`-algebra satisfies the equation

$(\forall F$ `: Flag) rev(rev(`F`)) =` F

we use the relation R, defined by $x\,R\,y$ iff either x and y are of sort `Bool` and $x = y$, or x and y are of sort `Flag` and $\texttt{up?}(x) = \texttt{up?}(y)$. To use coinduction, we need to show that:

- R is a behavioural congruence; i.e., it is preserved by the operations `up`, `dn`, and `rev`; and

- $\mathtt{rev}(\mathtt{rev}(x))\ R\ x$ for all x.

□

The proof obligations of this example are straightforward to demonstrate, and [10] use the relation R to show that all hidden `FLAG`-algebras satisfy the 'double reverse' equation.

3 Observational Truth

In hidden algebra, behavioural satisfaction of visible equations (i.e., equations whose left- and right-hand sides are of visible sort) is the same thing as standard satisfaction of equations. This is because the 'identity context', consisting of just the placeholder variable, is in $\mathsf{C}_\Sigma(v, v)$ for visible v. Moreover, if a hidden algebra satisfies an equation in the standard sense, then it behaviourally satisfies that equation. The notion of behavioural satisfaction is therefore based, in some sense, on the standard notion of satisfaction, and it is this sense that we aim to clarify in this section by showing that it is a *mode* of satisfaction.

We begin by distinguishing the objects that represent the results of observations:

Definition 4 An **observational basis** of a category C is a full subcategory $\mathsf{V} \hookrightarrow \mathsf{C}$ which is also discrete. □

Thus, an observational basis for C is a subcategory V where the only arrow (in C) going out of an object v in V is the identity arrow $1_v : v \to v$. The main idea here is to ignore any (algebraic) structure on the objects v in V: an observation simply produces some result. A similar notion of observational basis can be found in Burstall and Diaconescu [4] where a general method is given for constructing behavioural versions of logics, using institutions (as in Goguen and Burstall [7]).

Example 5 Given a hidden specification with no equations and with sorts $S = H \cup V$, where H contains all the hidden sorts and V the visible sorts, let C_Σ be the category with

objects: the sorts $s \in H + V$
arrows: $c : s \to t$ iff $c \in \mathsf{C}_\Sigma(s, t)$ (and if $s \in V$, then we require c to be the identity context).

Then V, as a discrete category, is an observational basis for C_Σ. □

Since this is the key motivating example, we use the following conventions: for an observational basis $\mathsf{V} \hookrightarrow \mathsf{C}$, we let s, t and u range over objects of C and v over objects of V; b and c range over arrows in C, and d ranges over arrows $s \to v$.

We are interested in *interpretations* of the category C; i.e., in functors $\mathsf{C} \to \mathsf{Set}$, sometimes referred to[1] as **presheaves** on C. For example, any hidden algebra gives rise to a presheaf A on C_Σ, As is the carrier set of sort $s \in S$, and $Ac : As \to At$ is the obvious interpretation of a context $c \in \mathsf{C}_\Sigma(s,t)$. However, not all presheaves F on C_Σ give rise to hidden algebras, because their interpretation Fv of visible sorts v is not constrained to be the fixed interpretation of hidden algebras.

It is well-known that categories of presheaves $\mathsf{Set}^{\mathsf{C}}$ form a topos (a useful introduction can be found in [19]; more comprehensive accounts in [17, 16]). For our purposes, the importance of this is that toposes have an 'internal' logical language whose 'truth values' are some Boolean algebra. In the case of presheaf categories such as $\mathsf{Set}^{\mathsf{C}}$, the truth values are sieves[2]:

Definition 6 For s in C, a **sieve on** s is a set of arrows with source s which is closed under postcomposition (i.e., if C is a sieve on s and $b : s \to t$, then for all $c : t \to u$ we have $b\,;c \in C$). We write Ωs for the set of sieves on s. Moreover, for $b : s \to t$ in C, defining $\Omega b : \Omega s \to \Omega t$ by

$$\Omega b C = \{\ c : t \to u \ \mid\ b\,;c \in C\ \}$$

makes $\Omega : \mathsf{C} \to \mathsf{Set}$ a presheaf. □

We now define observability as a modality on these truth values. When we view arrows in a sieve as 'paths to truth' [17], we add all the arrows that make a property observably true:

Definition 7 Define $\boxed{\circ} : \Omega \to \Omega$ by

$$\boxed{\circ}_s C = \{\ c : s \to t \ \mid\ (\forall v,\ d : t \to v)\, c\,;d \in C\ \}\ .$$

□

It is straightforward to check that $\boxed{\circ}$ is well-typed: i.e., that it takes sieves to sieves and that it is natural.

[1] Usually, a presheaf on C is taken to mean a functor $\mathsf{C}^{\mathrm{op}} \to \mathsf{Set}$; throughout this paper we follow a minority usage in taking this to mean simply a functor $\mathsf{C} \to \mathsf{Set}$

[2] More accurately, this is one way of viewing truth values; there are other views, but they are all isomorphic.

The characterisation of modalities in toposes seems to be due to Lawvere's axiomatisation of Grothendieck's generalised topologies, and the observation that these topological 'sites' could be viewed as a modality of 'local truth' [18]. The following is a standard definition (see, e.g., [2]):

Definition 8 A **modality** (in a topos) is an operator $j : \Omega \to \Omega$ such that

- $j\,\top = \top$,
- $j\,j\,\phi = j\,\phi$, and
- $j\,(\phi \wedge \psi) = j\,\phi \wedge j\,\psi$.

□

Fact 9 $\boxdot$ is a modality. In addition to the properties of modalities in the previous definition, it satisfies:

$$\boxdot(\phi \Rightarrow \psi) = \boxdot\phi \Rightarrow \boxdot\psi$$
$$\phi \Rightarrow \boxdot\phi$$

□

Thus, in any topos of presheaves on a category C with observational basis V, we can extend the internal logical language with a modality of observational truth. The Kripke-Beth-Joyal semantics of internal logical languages uses a relation of *forcing*:

Definition 10 Given $e : F \to G$ and $\phi : G \to \Omega$ in $\mathsf{Set}^{\mathsf{C}}$, we write $F \Vdash \phi(e)$ to indicate that ϕ is true 'at stage F'; i.e., that $e\,;\phi = !_F\,;\top$. The morphism e is referred to as a 'generalised element of G'.

We also write $F \Vdash_V \phi(e)$ to indicate visible truth at stage F; i.e., $e_v\,;\phi_v = !_{F\,v}\,;\top_v$ for all objects v in V. □

The following is a basic property of forcing that holds for any topos (see e.g., Proposition 8.2 of [16]); we use this as a lemma to extend the Kripke-Beth-Joyal semantics to our observational modality.

Property 11 If $G \Vdash \phi(e)$ and $f : F \to G$, then $F \Vdash \phi(f\,;e)$. □

Proposition 12 $F \Vdash \boxdot\phi(e)$ iff $F \Vdash_V \phi(e)$.

Proof. If $F \Vdash_V \phi(e)$ then for all v in V and $y \in F\,v$ we have $\phi_v\,(e_v\,y) = \{1_v\}$. In particular, given $d : s \to v$ and $x \in F\,s$, we take $y = F\,d\,x$ to get $\phi_v\,(e_v\,(F\,d\,x)) = \{1_v\}$, and naturality of $e\,;\phi$ means that $\Omega\,d(\phi_s\,(e_s\,x)) = \{1_v\}$, so $d \in \phi_s\,(e_s\,x)$. Since this holds for all d and x, it follows that $e\,;\phi\,;\boxdot = !_F\,;\top$, i.e., $F \Vdash \boxdot\phi(e)$.

Conversely, if $F \Vdash \boxdot\phi(e)$, then $e_v\,;\phi_v\,;\boxdot_v = !_{F\,v}\,;\top_v$ for all v in V, but $\boxdot_v$ is the identity on $\Omega\,v$ so $F \Vdash_V \phi(e)$. □

We can now give a general categorical version of the principle of coinduction described in Section 2:

Theorem 13 Let $e : F \to G$ and let $m : G_0 \to G$ be some arrow in $\mathsf{Set}^{\mathsf{C}}$ such that $G_0 \Vdash_V \phi(m)$; then $F \Vdash \boxdot\phi(e)$ whenever e factors through G_0.

Proof. Suppose $G_0 \Vdash_V \phi(m)$ and that e factors through G_0 as $e_0 \,;m$, then by Property 11 we have $F \Vdash_V \phi(e_0\,;m)$, i.e., $F \Vdash_V \phi(e)$, and it follows from Lemma 12 that $F \Vdash \boxdot\phi(e)$ as desired. □

The relationship with coinduction is as follows:

- G_0 is the 'candidate relation';
- $G \Vdash_V \phi(e)$ corresponds to the candidate relation being equality on visible sorts; and
- that G_0 is a subobject corresponds to the substitutive property of behavioural congruences.

A specialisation of the Kripke-Beth-Joyal semantics to presheaf categories uses the *homset* functors: for each object s in C we have $\mathsf{y}^s : \mathsf{C} \to \mathsf{Set}$, defined by $\mathsf{y}^s t = \mathsf{C}(s,t)$, the set of arrows from s to t in C, and for arrows $c : t \to u$, we have $\mathsf{y}^s t : \mathsf{y}^s u \to$ taking $b : s \to t$ to $b\,;c : s \to u$. Moreover, for $b : s' \to s$, we have $\mathsf{y}^b : \mathsf{y}^s \to \mathsf{y}^{s'}$, where for objects t of C and $c : s \to t$, $\mathsf{y}^b_t c = b\,;c : s' \to t$.

Notation 14 For $e : \mathsf{y}^s \to F$ and $\phi : F \to \Omega$, we write $s \Vdash \phi(e)$ for $\mathsf{y}^s \Vdash \phi(e)$. □

The significance of this notation is that natural transformations $\mathsf{y}^s : F \to$ correspond to elements of $F\,s$ (via the 'Yoneda embedding' where e corresponds to the element $e_s 1_s \in F\,s$), so the forcing notion of generalised element becomes the set theoretic notion of element.

The semantics of observational truth can be expressed in a form much closer to the spirit of the Kripke-Beth-Joyal semantics as follows:

Theorem 15 For $e : \mathsf{y}^s \to F$,

$$s \Vdash \boxdot\phi(e) \quad \text{iff} \quad \text{for all } d : s \to v \text{ in } \mathsf{C},\ \ v \Vdash \phi(e_d)\ ,$$

where $e_d = \mathsf{y}^d\,;e$.

Proof. Using Lemma 12, we have to show:

$$(1) \qquad \mathsf{y}^s \Vdash_V \phi(e) \quad \text{iff} \quad \text{for all } d : s \to v \text{ in } \mathsf{C},\ \ \mathsf{y}^v \Vdash_V \phi(e_d)\ .$$

Since $\mathsf{y}^v v = \mathsf{C}(v,v) = \{1_v\}$ and $e_d(1_v) = e_v(d)$, the right-hand side of this equivalence can be rephrased as:

$$\text{for all } d : s \to v \text{ in } \mathsf{C},\ \ \phi_v(e_v(d)) = \{1_v\}\ .$$

Moreover, since $\mathsf{y}^s v = \mathsf{C}(s,v)$, this is equivalent to $\mathsf{y}^s \Vdash_V \phi(e)$ which proves (1) as desired. $\square$

The principle of coinduction can now be reformulated as:

Corollary 16 Let $e : \mathsf{y}^s \to G$ and let $m : G_0 \to G$ be a subobject of G such that $G_0 \Vdash_V \phi(m)$; then $s \Vdash \boxdot\phi(e)$ whenever $e_s : \mathsf{y}^s s \to G_0\, s$. $\square$

4 Conclusion

We have presented presheaf categories that admit a notion of observational or behavioural truth as a modality, and extended the Kripke-Beth-Joyal semantics for presheaf toposes to this modality. This semantics allows a very natural categorical formulation of the principle of coinduction for the internal logics of these toposes.

The motivation for our notion of observational truth comes from hidden algebra, and hidden algebraic models provide examples of presheaves; however, not all the presheaves in our toposes give rise to hidden algebras. The reason for this is that hidden algebras have a fixed interpretation on observational bases, whereas the presheaves we consider do not. One avenue for further exploration is to more closely relate hidden algebras and the presheaves considered here, perhaps by looking at the fibrational structure of these toposes, where each fibre has a fixed interpretation of the observational basis.

References

[1] Peter Aczel and Nax Mendler. A final coalgebra theorem. In D.H. Pitt et al., editor, *Category Theory and Computer SCience*. Springer-Verlag Lecture Notes in Computer Science 389, 1989.

[2] John L. Bell. *Toposes and Local Set Theories: an introduction*. Number 14 in Oxford Logic Guides. Oxford Science Publications, 1988.

[3] Michel Bidoit, Rolf Hennicker, and Martin Wirsing. Behavioural and abstractor specifications. *Science of Computer Programming*, 25(2–3):149–186, 1995.

[4] Rod Burstall and Răzvan Diaconescu. Hiding and behaviour: an institutional approach. In A. W. Roscoe, editor, *A Classical Mind: essays dedicated to C.A.R. Hoare*. Prentice-Hall International, 1994.

[5] Silvio Ghilardi and Giancarlo Meloni. Modal and tense predicate logic: models in presheaves and categorical conceptualisation. In Francis Borceux, editor, *Categorical Algebra and its Applications*, volume 1348 of *Springer Lecture Notes in Mathematics*, pages 130–142. Springer-Verlag, Heidelberg, 1988.

[6] Joseph A. Goguen. Types as theories. In George Michael Reed, Andrew William Roscoe, and Ralph F. Wachter, editors, *Topology and Category Theory in Computer Science*, pages 357–390. Oxford University Press, 1991.

[7] Joseph A. Goguen and Rod Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, 1992.

[8] Joseph A. Goguen and Kai Lin. Web-based support for cooperative software engineering. *Annals of Software Engineering*, 12, 2001.

[9] Joseph A. Goguen and Grant Malcolm. Hidden coinduction: behavioral correctness proofs for objects. *Mathematical Structures in Computer Science*, 9(3):287–320, 1999.

[10] Joseph A. Goguen and Grant Malcolm. A hidden agenda. *Theoretical Computer Science*, 245(1):55–101, 2000.

[11] Robert Goldblatt. Grothendieck topology as geometric modality. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 27:495–529, 1981. Reprinted in Goldblatt, *Mathematics of Modality*, CSLI publications, 1983.

[12] Rolf Henniker. Context induction: a proof principle for behavioural abstractions. In A. Miola, editor, *Design and Implementation of Symbolic Computation Systems*. Springer-Verlag Lecture Notes in Computer Science 429, 1990.

[13] B. Jacobs. Objects and classes, co-algebraically. In B. Freitag, C.B. Jones, C. Lengauer, and H.-J. Schek, editors, *Object-Orientation with Parallelism and Persistence*, pages 83–103. Kluwer Acad. Publ., 1996.

[14] Bart Jacobs and Jan J.M.J. Rutten. A tutorial on (co)algebras and (co)induction. *Bulletin of the European Association for Theoretical Computer Science*, 62:222–259, 1997.

[15] Peter Johnstone, John Power, Toru Tsujishita, Hiroshi Watanabe, and James Worrell. An axiomatics for categories of transition systems as coalgebras. In *Proc. Logic in Computer Science*, pages 207–213, 1998.

[16] Joachim Lambek and Philip J. Scott. *Introduction to Higher Order Categorical Logic*. Cambridge University Press, 1986. Cambridge Studies in Advanced Mathematics, Volume 7.

[17] Saunders Mac Lane and Ieke Moerdijk. *Sheaves in Geometry and Logic*. Springer-Verlag, 1992.

[18] F. William Lawvere. Quantifiers and sheaves. In *Actes du Congrès International des Mathématiciens*, pages 329–334. Gauthier-Villars, 1971.

[19] F. William Lawvere and Stephen H. Schanuel. *Conceptual Mathematics*. Cambridge University Press, 1997.

[20] Grant Malcolm. Behavioural equivalence, bisimilarity, and minimal realisation. In Magne Haveraaen, Olaf Owe, and Ole-Johan Dahl, editors, *Recent Trends in Data Type Specifications. 11th Workshop on Specification of Abstract Data Types, WADT11. Oslo Norway, September 1995*, pages 359–378. Springer-Verlag Lecture Notes in Computer Science 1130, 1996.

[21] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, Parts I and II. Technical Report ECS-LFCS-89-85 and -86, Department of Computer Science, Edinburgh University, 1989.

[22] Robin Milner and Mads Tofte. Co-induction in relational semantics. *Theoretical Computer Science*, 87(1):209–220, 1991.

[23] Fernando Orejas, Marisa Navarro, and Ana Sánchez. Implementation and behavioural equivalence: a survey. In M. Bidoit and C. Choppy, editors, *Recent Trends in Data Type Specification*. Springer-Verlag Lecture Notes in Computer Science 655, 1993.

[24] Horst Reichel. Behavioural equivalence – a unifying concept for initial and final specifications. In *Proceedings, Third Hungarian Computer Science Conference*. Akademiai Kiado, 1981. Budapest.

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

[26] Donald Sannella and Andrzej Tarlecki. Toward formal development of programs from algebraic specifications. *Acta Informatica*, 25:233–281, 1988.

[27] James Worrell. Toposes of coalgebras and hidden algebras. In Bart Jacobs, Larry Moss, Horst Reichel, and Jan Rutten, editors, *Proc. Coalgebraic Methods in Computer Science 1998*, volume 11 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2000.