Mathematical Structures in Computer Science (2022), **32**, pp. 374–419
doi:10.1017/S0960129521000219

CAMBRIDGE UNIVERSITY PRESS

PAPER

The costructure–cosemantics adjunction for comodels for computational effects

Richard Garner*

Department of Mathematics, Macquarie University, NSW 2109, Australia
*Corresponding author. Email: richard.garner@mq.edu.au

(Received 29 November 2020; revised 9 July 2021; accepted 12 August 2021; first published online 6 December 2021)

Abstract

It is well established that equational algebraic theories and the monads they generate can be used to encode computational effects. An important insight of Power and Shkaravska is that *comodels* of an algebraic theory $\mathbb{T}$ – i.e., models in the opposite category $\mathcal{S}\mathrm{et}^{\mathrm{op}}$ – provide a suitable environment for evaluating the computational effects encoded by $\mathbb{T}$. As already noted by Power and Shkaravska, taking comodels yields a functor from accessible monads to accessible comonads on $\mathcal{S}\mathrm{et}$. In this paper, we show that this functor is part of an adjunction – the "costructure–cosemantics adjunction" of the title – and undertake a thorough investigation of its properties. We show that, on the one hand, the cosemantics functor takes its image in what we term the *presheaf comonads* induced by small categories; and that, on the other, costructure takes its image in the *presheaf monads* induced by small categories. In particular, the cosemantics comonad of an accessible monad will be induced by an explicitly-described category called its *behaviour category* that encodes the static and dynamic properties of the comodels. Similarly, the costructure monad of an accessible comonad will be induced by a behaviour category encoding static and dynamic properties of the comonad coalgebras. We tie these results together by showing that the costructure–cosemantics adjunction is *idempotent*, with fixpoints to either side given precisely by the presheaf monads and comonads. Along the way, we illustrate the value of our results with numerous examples drawn from computation and mathematics.

1. Introduction

It is a pleasure to contribute to this festschrift volume for John Power. I first met John about fifteen years ago during my time as a PhD student in Cambridge, when in particular, he served as my external thesis examiner: an encounter which could fairly be described as "character-building". Since then, our paths have crossed all too infrequently – that is, until recently, when John once and for all put the dark British winter days behind him and came home to Australia. Since then, John has once again become a regular feature of the Australian Category Seminar, and through his invigorating talks, I have come to appreciate more deeply the interconnections between his category-theoretic contributions in the Australian style and their applications in the world of computer science. This article is in that spirit and revolves around some of John's perspicuous contributions to the study of computational effects and their semantics.

It is a well-known story that the category-theoretic approach to computational effects originates with Moggi [25]. Given a cartesian closed category $\mathcal{C}$, providing a denotational semantics for a base notion of computation, this approach allows additional language features such as input/output, interaction with the store, or non-determinism – all falling under the general rubric of "effects" – to be encoded in terms of algebraic structure borne by objects of $\mathcal{C}$.

In Moggi's treatment, this structure is specified via a strong monad T on $\mathcal{C}$; a disadvantage of this approach is that, in taking as primitive the objects $T(A)$ of *computations* with effects from T and values in A, it gives no indication of how the effects involved are to be encoded as language features. An important thread [28–30] in John's work with Gordon Plotkin has sought to rectify this, by identifying a computational effect not with a strong monad *per se*, but rather with a set of (computationally meaningful) algebraic operations and equations that *generate* a strong monad, in the sense made precise by John and Max Kelly in [20].

The most elementary case of the above takes $\mathcal{C} = \mathcal{S}\mathrm{et}$: then a computational effect in Moggi's sense is simply a monad on $\mathcal{S}\mathrm{et}$, while an effect in the Plotkin–Power sense is an equational *algebraic theory*, involving a signature of (possibly infinitary) operations and a set of equations between terms in the signature. As the same monad may admit many different presentations, the Plotkin–Power approach is more refined, but it is also slightly narrower in scope, as not every monad on $\mathcal{S}\mathrm{et}$ is engendered by an algebraic theory, but only the *accessible* ones – also called *monads with rank*. (A well-known inaccessible monad is the continuation monad $V^{V^{(-)}}$).

The Plotkin–Power approach makes it particularly easy to define *models* in any category $\mathcal{A}$ with powers: they are $\mathcal{A}$-objects endowed with interpretations of the given operations which satisfy the given equations. When $\mathcal{A} = \mathcal{S}\mathrm{et}$, such models are the same as algebras for the associated monad; computationally, these can be interpreted as sets of effectful computations which have been identified up to a notion of equivalence which respects the effect semantics.

It is another important insight of John, in collaboration with Olha Shkaravska [32], that there is also a computational interpretation of *comodels*. A comodel in $\mathcal{A}$ is simply a model in $\mathcal{A}^{\mathrm{op}}$, and again an important case is where $\mathcal{A} = \mathcal{S}\mathrm{et}$. The idea is that, given an algebraic theory $\mathbb{T}$ for effects which interact with an "environment"[1], a comodel of $\mathbb{T}$ in $\mathcal{S}\mathrm{et}$ provides the kind of environment with which such programs interact. The underlying set S of such a comodel is the set of possible states of the environment; while each generating A-ary operation σ of the theory – which requests an element of A from the environment and binds it – is co-interpreted as a function $(\llbracket\sigma\rrbracket\colon S \to A \times S)$ which answers the request, and moves to a new state. While, in the first instance, we co-interpret only the generating $\mathbb{T}$-operations, we can extend this inductively to all A-ary computations $t \in T(A)$ of the associated monad; whereupon we can see the co-interpretation $\llbracket t\rrbracket\colon S \to A \times S$ as a way of *running* (c.f. [37]) the computation $t \in T(A)$ starting from some state $s \in S$ to obtain a return value in A and a final state in S.

The computational perspective on comodels is powerful and has achieved significant traction in computer science; see, for example, [4, 24, 26, 27, 31, 37]. However, our objective in this paper is to return to the original [32] and settle some of the unanswered questions posed there. Power and Shkaravska observe that the category of comodels of any algebraic theory is *comonadic* over $\mathcal{S}\mathrm{et}$, so that, if we choose to identify algebraic theories with monads, then we have a process which associates to each accessible monad on $\mathcal{S}\mathrm{et}$ a certain comonad on $\mathcal{S}\mathrm{et}$. This leads them to ask: "Does the construction of a comonad on $\mathcal{S}\mathrm{et}$ from a monad with rank on $\mathcal{S}\mathrm{et}$ yield an interesting relationship between monads and comonads?"

We will answer this question in the affirmative, by showing that this construction provides the right adjoint part of a dual adjunction

$$\mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}} \underset{\text{Cosem}}{\overset{\text{Costr}}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et}) \tag{1}$$

between the categories of accessible monads and accessible comonads on $\mathcal{S}\mathrm{et}$. The corresponding left adjoint can be seen as taking a comonad Q on $\mathcal{S}\mathrm{et}$ to the largest algebraic theory for which any Q-coalgebra is a comodel. Following [11, 22], we refer to the two functors in this adjunction as "cosemantics" and "costructure".

[1] We use this word in its everyday sense, with no particular technical meaning attached.

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

In fact, the mere existence of the adjunction (1) is not hard to establish. Indeed, as we will see, its two directions were already described in [19], with our costructure corresponding to their *dual monad* of a comonad, and our cosemantics being their *Sweedler dual comonad* of a monad. Our real contribution is that we do not seek merely to construct (1), but also to understand it thoroughly and concretely.

In one direction, we will explicitly calculate the cosemantics functor, this will, among other things, answer [32]'s request that "we should very much like to be able to characterise those comonads, at least on $\mathcal{S}\mathrm{et}$, that arise from categories of comodels". These comonads will be what we term *presheaf comonads*; for a small category $\mathbb{B}$, the associated presheaf comonad $\mathsf{Q}_{\mathbb{B}}$ is that induced by the adjunction

$$\mathcal{S}\mathrm{et}^{\mathbb{B}} \underset{\mathrm{res}_J}{\overset{\mathrm{ran}_J}{\underset{\longrightarrow}{\longleftarrow}}} \mathcal{S}\mathrm{et}^{\mathrm{ob}(\mathbb{B})} \underset{\Sigma}{\overset{\Delta}{\underset{\longrightarrow}{\longleftarrow}}} \mathcal{S}\mathrm{et} \quad .$$

whose left part is restriction and right Kan extension along the inclusion-of-objects functor $J\colon \mathrm{ob}(\mathbb{B}) \to \mathbb{B}$; more explicitly, we have that $Q_{\mathbb{B}}(A) = \sum_{b\in\mathbb{B}} \prod_{c\in\mathbb{B}} A^{\mathbb{B}(b,c)}$. Comonads of this form are known entities in computer science: they are precisely the interpretations of *directed containers* as introduced in [5], and in [8] were termed *dependently typed coupdate comonads*.

In fact, we do more than merely characterising the image of the cosemantics functor: we prove for each accessible monad T on $\mathcal{S}\mathrm{et}$ that its image under cosemantics is the presheaf comonad of an *explicitly* given category $\mathbb{B}_{\mathsf{T}}$, which we term the *behaviour category* of T. Since the category of Eilenberg–Moore $\mathsf{Q}_{\mathbb{B}_{\mathsf{T}}}$-coalgebras is equivalent to the functor category $[\mathbb{B}_{\mathsf{T}}, \mathcal{S}\mathrm{et}]$, we may also state this result as:

Theorem. *Given an accessible monad* T *with behaviour category* $\mathbb{B}_{\mathsf{T}}$*, the category of* T*-comodels is equivalent to* $[\mathbb{B}_{\mathsf{T}}, \mathcal{S}\mathrm{et}]$ *via an equivalence commuting with the forgetful functors to* $\mathcal{S}\mathrm{et}$*.*

Our description of the behaviour category $\mathbb{B}_{\mathsf{T}}$ is quite intuitive. Objects $\beta \in \mathbb{B}_{\mathsf{T}}$ are elements of the final T-comodel in $\mathcal{S}\mathrm{et}$, which may be described in many ways; we give a novel presentation as what we term *admissible behaviours* of T. These comprise families $(\beta_A\colon T(A) \to A)_{A\in\mathcal{S}\mathrm{et}}$ of functions, satisfying axioms expressing that β acts like a state of a comodel in providing a uniform way of running T-computations to obtain values. As for morphisms of $\mathbb{B}_{\mathsf{T}}$, these will be transitions between admissible behaviours determined by T-*commands*, i.e., unary operations $m \in T(1)$. We will see that maps with domain β in $\mathbb{B}_{\mathsf{T}}$ are T-commands identified up to an equivalence relation $\sim_\beta$ which identifies commands which act in the same way on all states of behaviour β.

We also describe the action of the cosemantics functor on morphisms. Thus, given a map of monads $f\colon \mathsf{T}_1 \to \mathsf{T}_2$ – which encodes an *interpretation* or *compilation* of effects – we describe the induced map of presheaf comonads $\mathsf{Q}_{\mathbb{B}_{\mathsf{T}_2}} \to \mathsf{Q}_{\mathbb{B}_{\mathsf{T}_1}}$. As explained in [8], maps of presheaf comonads do not correspond to functors, but rather to so-called *cofunctors* [3, 16] of the corresponding categories, involving a mapping *forwards* at the level of objects, and mappings *backwards* on morphisms. We are able to give an explicit description of the cofunctor on behaviour categories induced by a monad morphism.

Theorem. *The functor* $\mathbb{B}_{(-)}\colon \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}} \to \mathcal{C}\mathrm{of}$ *taking each accessible monad to its behaviour category, and each map of accessible monads to the induced cofunctor on behaviour categories yields a within-isomorphism factorisation of the cosemantics functor through the category* $\mathcal{C}\mathrm{at}$ *of small categories and cofunctors:*

$$\begin{array}{ccc} & & \mathcal{C}\mathrm{of} \\ & \overset{\mathbb{B}_{(-)}}{\nearrow} & \big\downarrow \mathsf{Q}_{(-)} \\ \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}} & \xrightarrow{\mathrm{Cosem}} & \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et}) \end{array} .$$

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

For the other direction of the adjunction (1), we will in an analogous manner give an explicit calculation of the image of the costructure functor. On objects, the monads in this image are what we term *presheaf monads*; here, for a small category $\mathbb{B}$, the presheaf monad $\mathsf{T}_{\mathbb{B}}$ is that induced by the adjunction:

$$\mathcal{S}\mathrm{et}^{\mathbb{B}^{\mathrm{op}}} \underset{\mathrm{res}_J}{\overset{\mathrm{lan}_J}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathcal{S}\mathrm{et}^{\mathrm{ob}(\mathbb{B})} \underset{\Pi}{\overset{\Delta}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathcal{S}\mathrm{et} \ ,$$

with the explicit formula $T_{\mathbb{B}}(A) = \prod_{b \in \mathbb{B}} \sum_{c \in \mathbb{B}} \mathbb{B}(b,c) \times A$ (note the duality with the comonad $\mathsf{Q}_{\mathbb{B}}$). Much like before, we will prove for each accessible comonad Q on $\mathcal{S}\mathrm{et}$ that its image under costructure is of the form $\mathsf{T}_{\mathbb{B}_{\mathsf{Q}}}$ for an explicitly given "behaviour category" $\mathbb{B}_{\mathsf{Q}}$. The picture is perhaps less compelling in this direction, but the objects of $\mathbb{B}_{\mathsf{Q}}$ can again be described as "behaviours", by which we now mean elements of the final Q-coalgebra $Q(1)$; while morphisms of $\mathbb{B}_{\mathsf{Q}}$ are uniform ways of transitioning between Q-behaviours. Like before, we also compute the costructure functor on morphisms and again find that each comonad morphism induces a cofunctor between behaviour categories, so yielding our third main result:

Theorem. *The functor* $\mathbb{B}_{(-)} \colon \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et}) \to \mathcal{C}\mathrm{of}$ *taking an accessible comonad to its behaviour category, and a map of accessible comonads to the induced cofunctor on behaviour categories yields a within-isomorphism factorisation*

$$\begin{array}{ccc} & & \mathcal{C}\mathrm{of} \\ & \overset{\mathbb{B}_{(-)}}{\nearrow} & \big\downarrow \mathsf{Q}_{(-)} \\ \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et}) & \xrightarrow{\mathrm{Costr}} & \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}} \ . \end{array}$$

It remains only to understand how costructure and cosemantics interact with each other. The crucial observation is that (1) is an example of a so-called *idempotent* (or *Galois*) adjunction. Here, an adjunction $F \dashv G \colon \mathcal{D} \to \mathcal{C}$ is *idempotent* if any application of F yields a *fixpoint to the left*, i.e., an object of $\mathcal{D}$ at which the adjunction counit is invertible, while any application of G yields a fixpoint to the right, i.e., an object of $\mathcal{C}$ at which the adjunction unit is invertible. In these terms, our final main result can be stated as:

Theorem. *The costructure–cosemantics adjunction (1) is idempotent. Its fixpoints to the left and the right are the presheaf monads and presheaf comonads.*

Let us note that the results of this paper are only the first step in a larger investigation. On the one hand, to deal with recursion, we will require a comprehensive understanding of *enriched* versions of the costructure–cosemantics adjunction. On the other hand, even in the unenriched world of equational algebraic theories, we may be interested in understanding costructure and cosemantics for comodels in other categories than $\mathcal{S}\mathrm{et}$: for example, *topological* comodels, which encode information not only about behaviours of states but also about finitistic, computable observations of such behaviour. We hope to pursue these avenues in future work.

We conclude this introduction with a brief overview of the contents of the paper. We begin in Section 2 with background material on algebraic theories, their models and comodels and the relation to monads on $\mathcal{S}\mathrm{et}$, along with relevant examples relating to computational effects. In Section 3, we prepare the ground for our main results by investigating the classes of presheaf monads and comonads. Then in Section 4, we give the construction of the costructure–cosemantics adjunction (1), and explain how its two directions encapsulate constructions of [19].

In Section 5, we calculate the values of the cosemantics functor. We begin with a general category-theoretic argument that shows that its must take values in presheaf comonads; we then give a concrete calculation of the presheaf comonad associated with a given accessible monad, or in other words, of the behaviour category of the given monad. We also describe the cofunctors between behaviour categories induced by monad morphisms.

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

In Section 6, we turn to the costructure functor, showing by a direct calculation that it sends each accessible comonad to the presheaf monad of an appropriate behaviour category. As before, we also describe the cofunctor on behaviour categories induced by each map of accessible comonads. Then in Section 7, we tie these results together by exhibiting the idempotency of the costructure–cosemantics adjunction and characterising its fixpoints as the presheaf monads and comonads. Finally, Sections 8 and 9 are devoted to examples of behaviour categories and cofunctors calculated using our main results.

2. Algebraic Theories and Their (Co)models

2.1. Algebraic theories

In this background section, we recall the definition of (possibly infinitary) algebraic theory; the notions of model and comodel in any suitable category; and the relation to monads on $\mathcal{S}\mathrm{et}$. We also recall the applications of these notions in the study of computational effects.

Definition 2.1 (Algebraic theory). *A* signature *comprises a set* Σ *of* function symbols, *and for each* $\sigma \in \Sigma$ *a set* $|\sigma|$, *its* arity. *Given a signature* Σ *and a set* A, *we define the set* $\Sigma(A)$ *of* Σ-terms with variables in A *by the inductive clauses*

$$a \in A \implies a \in \Sigma(A) \quad \text{and} \quad \sigma \in \Sigma, t \in \Sigma(A)^{|\sigma|} \implies \sigma(t) \in \Sigma(A)\,.$$

An equation *over a signature* Σ *is a triple* (A,t,u) *with* A *a set and* $t, u \in \Sigma(A)$. *An* algebraic theory $\mathbb{T}$ *is a signature* Σ *and a set* $\mathcal{E}$ *of equations over it.*

Definition 2.2 ($\mathbb{T}$-terms). *Given a signature* Σ *and terms* $t \in \Sigma(A)$ *and* $u \in \Sigma(B)^A$, *we define the* substitution $t(u) \in \Sigma(B)$ *recursively by*

$$a \in A \implies a(u) = u_a \quad \text{and} \quad \sigma \in \Sigma, t \in \Sigma(A)^{|\sigma|} \implies (\sigma(t))(u) = \sigma(\lambda i.\, t_i(u))\,.$$

Given an algebraic theory $\mathbb{T} = (\Sigma, \mathcal{E})$ *and a set* B, *we define* $\mathbb{T}$-equivalence *to be the smallest equivalence relation* $\equiv_{\mathbb{T}}$ *on* $\Sigma(B)$ *such that:*

1. *If* $(A, t, u) \in \mathcal{E}$ *and* $v \in \Sigma(B)^A$, *then* $t(v) \equiv_{\mathbb{T}} u(v)$;
2. *If* $\sigma \in \Sigma$ *and* $t_i \equiv_{\mathbb{T}} u_i$ *for all* $i \in |\sigma|$, *then* $\sigma(t) \equiv_{\mathbb{T}} \sigma(u)$.

The set $T(A)$ *of* $\mathbb{T}$*-terms with variables in* A *is the quotient* $\Sigma(A)/\equiv_{\mathbb{T}}$.

When an algebraic theory $\mathbb{T}$ is thought of as specifying a computational effect, we think of $T(A)$ as giving the set of computations with effects from $\mathbb{T}$ and returning a value in A. The following standard examples illustrate this.

Example 2.3 (Input). *Given a set* V, *the theory of* V-valued input *comprises a single* V*-ary function symbol* read, *satisfying no equations, whose action we think of as:*

$$(t\colon V \to X) \mapsto \mathsf{let\ read()\ be}\ v.\ t(v)\,.$$

For this theory, terms $t \in T(A)$ *are computations that can request* V*-values from an external source and use them to determine a return value in* A. *For example, when* $V = \mathbb{N}$, *the program which requests two input values and returns their sum is encoded by*

$$\mathsf{let\ read()\ be}\ n.\ \mathsf{let\ read()\ be}\ m.\ n + m\ \in\ T(\mathbb{N})\,. \tag{2}$$

For an algebraic theory *qua* computational effect, it is idiomatic that its function symbols are read in continuation-passing style, so the domain of a function symbol $X^V \to X$ is a scope in

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

which an element of V is available to determine a continuation, and applying the operation binds this element to yield a continuation *simpliciter*.

Example 2.4 (Output). *Given a set V, the theory of* V-valued output *comprises an V-indexed family of unary function symbols* $(\mathsf{write}_v : v \in V)$ *subject to no equations. In the continuation-passing style, we denote the action of* write_v *by*

$$t \mapsto \mathsf{let}\ \mathsf{write}(v)\ \mathsf{be}\ \_\,.\,t \qquad \text{or, more simply,} \qquad t \mapsto \mathsf{write}(v); t\,.$$

Example 2.5 (Read-only state). *Given a set V, the theory of* V-valued read-only state *has a single V-ary operation* get*, satisfying the equations*

$$\mathsf{get}(\lambda v.\, x) \equiv x \qquad \text{and} \qquad \mathsf{get}(\lambda v.\, \mathsf{get}(\lambda w.\, x_{vw})) \equiv \mathsf{get}(\lambda v.\, x_{vv})\,. \tag{3}$$

These equations express that reading from read-only state should not change that state, and that repeatedly reading the state should always yield the same answer; in another nomenclature, these axioms express that get *is* copyable *and* discardable *[36]. Note also that if V is a two-element set, and we write* $\mathsf{get}(x, y)$ *as $x \cdot y$, then these equations become $x \cdot x \equiv x$ and $(x \cdot y) \cdot (w \cdot z) \equiv x \cdot z$, which are easily seen to be equivalent to the equations asserting that $\cdot$ is an idempotent associative operation satisfying the "rectangular band" identity $xyx \equiv x$.*

Example 2.6 (State, [29]). *Given a set V, the theory of* V-valued state *comprises an V-ary operation* get *and a V-indexed family of unary operations* put_v*, subject to the following equations:*

$$\mathsf{get}(\lambda v.\, \mathsf{put}_v(x)) \equiv x \quad \mathsf{put}_u(\mathsf{put}_v(x)) \equiv \mathsf{put}_v(x) \quad \mathsf{put}_u(\mathsf{get}(\lambda v.\, x_v)) \equiv \mathsf{put}_u(x_u)\,.$$

Read in continuation-passing style, these axioms capture the semantics of reading and updating a store containing an element of V.

We now describe the appropriate morphisms between algebraic theories.

Definition 2.7 (Category of algebraic theories). *Let $\mathsf{T}_1 = (\Sigma_1, \mathcal{E}_1)$ and $\mathsf{T}_2 = (\Sigma_2, \mathcal{E}_2)$ be algebraic theories. An* interpretation $f\colon \mathbb{T}_1 \to \mathbb{T}_2$ *is given by specifying, for each $\sigma \in \Sigma_1$, a term $\sigma^f \in \Sigma_2(|\sigma|)$ such that, on defining for each $t \in \Sigma_1(A)$ the term $t^f \in \Sigma_2(A)$ by the recursive clauses*

$$a \in A \Rightarrow a^f = a \quad \text{and} \quad \sigma \in \Sigma, t \in \Sigma(A)^{|\sigma|} \Rightarrow (\sigma(t))^f = \sigma^f(\lambda a.\, (t_a)^f)\,, \tag{4}$$

we have that $t^f \equiv_{\mathbb{T}_2} u^f$ for all $(A, t, u) \in \mathcal{E}_1$. With the obvious composition, we obtain a category $\mathcal{A}\mathrm{lg}\mathcal{T}\mathrm{h}$ algebraic theories and interpretations.

An interpretation $\mathbb{T}_1 \to \mathbb{T}_2$ between theories can be understood as a way of translating computations with effects from $\mathbb{T}_1$ into ones with effects from $\mathbb{T}_2$.

Example 2.8 *Let $h\colon V \to W$ be a function between sets, let $\mathbb{T}_1$ be the theory of V-valued output, and let $\mathbb{T}_2$ be the theory of W-valued state. We have an interpretation $f\colon \mathbb{T}_1 \to \mathbb{T}_2$ defined by* $\mathsf{write}_v^f = \mathsf{put}_{h(v)}$.

Example 2.9 *Let $h\colon W \to V$ be a function between sets, let $\mathbb{T}_1$ be the theory of V-valued read-only state, and let $\mathbb{T}_2$ be the theory of W-valued state. We have an interpretation $f\colon \mathbb{T}_1 \to \mathbb{T}_2$ defined by* $\mathsf{get}^f = \mathsf{get}(\lambda w.\, h(w))$.

2.2. Models and comodels

We now describe the notions of *model* and *comodel* for an algebraic theory. We begin by establishing some necessary notation.

Notation 2.10 We say that a category $\mathcal{C}$ has *powers* if, for every $X \in \mathcal{C}$ and set A, an A-fold self-product $(\pi_a\colon X^A \to X)_{a \in A}$ exists in $\mathcal{C}$. Given an A-indexed family of maps $f_a\colon Y \to X$, we write $(f_a)_{a \in A}$ for the unique map $Y \to X^A$ such that $\pi_b \circ (f_a)_{a \in A} = f_b$ for all $b \in A$. Dually, we say $\mathcal{C}$

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

has *copowers* if all A-fold self-coproducts $(\iota_a : X \to A \cdot X)_{a \in A}$ exist in $\mathcal{C}$; and given an A-indexed family of maps $f_a : X \to Y$, we write $\langle f_a \rangle_{a \in A}$ for the unique map $A \cdot X \to Y$ with $\langle f_a \rangle_{a \in A} \circ \iota_b = f_b$.

Definition 2.11 (Σ-structure). *Let Σ be a signature. A Σ-structure $\boldsymbol{X} = (X, [\![-]\!]_{\boldsymbol{X}})$ in a category $\mathcal{C}$ with powers comprises an underlying object $X \in \mathcal{C}$ and operations $[\![\sigma]\!]_{\boldsymbol{X}} : X^{|\sigma|} \to X$ for each $\sigma \in \Sigma$. Given a Σ-structure $\boldsymbol{X} \in \mathcal{C}^{\Sigma}$, we define for each $t \in \Sigma(A)$ the derived operation $[\![t]\!]_{\boldsymbol{X}} : X^A \to X$ by the following recursive clauses:*

$$[\![a]\!]_{\boldsymbol{X}} = \pi_a \qquad \text{and} \qquad [\![\sigma(t)]\!]_{\boldsymbol{X}} = X^A \xrightarrow{([\![t_i]\!]_{\boldsymbol{X}})_{i \in |\sigma|}} X^{|\sigma|} \xrightarrow{[\![\sigma]\!]_{\boldsymbol{X}}} X\,. \tag{5}$$

Definition 2.12 ($\mathbb{T}$-model). *Let $\mathbb{T} = (\Sigma, \mathcal{E})$ be an algebraic theory. A $\mathbb{T}$-model in a category with powers $\mathcal{C}$ is a Σ-structure $\boldsymbol{X}$ such that $[\![t]\!]_{\boldsymbol{X}} = [\![u]\!]_{\boldsymbol{X}} : X^A \to X$ for all $(A, t, u) \in \mathcal{E}$. We write $\mathcal{C}^{\mathbb{T}}$ for the category whose objects are $\mathbb{T}$-models in $\mathcal{C}$, and whose maps $\boldsymbol{X} \to \boldsymbol{Y}$ are $\mathcal{C}$-maps $f : X \to Y$ such that $[\![\sigma]\!]_{\boldsymbol{Y}} \circ f^{|\sigma|} = f \circ [\![\sigma]\!]_{\boldsymbol{X}}$ for all $\sigma \in \Sigma$. We write $U^{\mathbb{T}} : \mathcal{C}^{\mathbb{T}} \to \mathcal{C}$ for the obvious forgetful functor.*

Lemma 2.13 (Substitution, soundness). *If $\boldsymbol{X}$ is a $\mathbb{T}$-model in $\mathcal{C}$, then:*

(i) For all $t \in T(B)$ and $u \in T(A)^B$ we have that

$$[\![t(u)]\!]_{\boldsymbol{X}} = X^A \xrightarrow{([\![u_b]\!]_{\boldsymbol{X}})_{b \in B}} X^B \xrightarrow{[\![t]\!]} X\,;$$

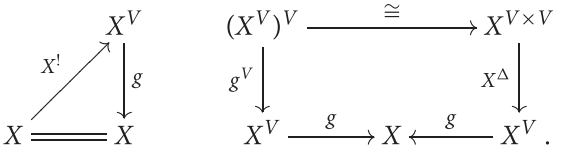
(ii) If $t \equiv_{\mathbb{T}} u$ in $T(A)$, then $[\![t]\!]_{\boldsymbol{X}} = [\![u]\!]_{\boldsymbol{X}} : X^A \to X$.

Example 2.14 *Given an object X of a category $\mathcal{C}$ with powers, we see that:*

(i) To make X a model of the theory of V-valued output is to endow it with a V-indexed family of maps $X \to X$, or equally, a single map $X \to X^V$.

(ii) To make X a model of the theory of V-valued input is to endow it with a map $X^V \to X$.

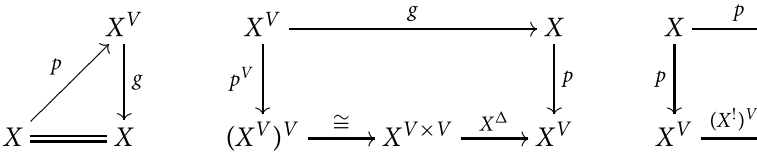
(iii) To make X a model of V-valued read-only state is to give a map $g : X^V \to X$ which renders commutative the diagrams



When $\mathcal{C} = \mathbb{S}\mathrm{et}$, [17, Proposition 4.3] shows that the category of such models is equivalent to the full subcategory of $\mathbb{S}\mathrm{et}^V$ whose objects are families of sets $(X_v : v \in V)$ which are either all empty or all non-empty. The model corresponding to such a family is given by (X, g) where $X = \prod_{v \in V} X_v$ and

$$g : (\prod_{v \in V} X_v)^V \to \prod_{v \in V} X_v \qquad \lambda v.\, (\lambda w.\, x_{vw}) \mapsto \lambda v.\, x_{vv}\,. \tag{6}$$

(iv) To make X into a model of the theory of V-valued state is to give maps $g : X^V \to X$ and $p : X \to X^V$ which render commutative the diagrams



The category of such models in $\mathbb{S}\mathrm{et}$ is in fact equivalent to $\mathbb{S}\mathrm{et}$ itself; a more detailed analysis shows that it is isomorphic to the category whose objects are families of sets $(X_v : v \in$

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

V) endowed with functions (necessarily isomorphisms) $\theta_{vw} \colon X_v \to X_w$ *satisfying* $\theta_{vv} = \mathrm{id}_{X_v}$ *and* $\theta_{vw}\theta_{uv} = \theta_{uw}$. *The model corresponding to such an object has underlying object* $X = \prod_{v \in V} X_v$, *with g given as in (6), and with*

$$p \colon \prod_{v \in V} X_v \to \big(\prod_{v \in V} X_v\big)^V \qquad \lambda v.\, x_v \mapsto \lambda v.\, \lambda w.\, \theta_{vw}(x_v)\,.$$

Dual to the notion of model is the notion of *comodel*. Recall that a category $\mathcal{C}$ has *copowers* if every A-fold self-coproduct $(\nu_a \colon X \to A \cdot X)_{a \in A}$ exists in $\mathcal{C}$.

Definition 2.15 (Comodel). *Let* $\mathbb{T}$ *be an algebraic theory. A* $\mathbb{T}$-comodel *in a category* $\mathcal{C}$ *with copowers is a model of* $\mathbb{T}$ *in* $\mathcal{C}^{\mathrm{op}}$; *it thus comprises an object* $X \in \mathcal{C}$ *and "co-operations"* $[\![\sigma]\!]_X \colon X \to |\sigma| \cdot X$, *subject to the equations of* $\mathbb{T}$. *We write* ${}^{\mathbb{T}}\mathcal{C}$ *for the category of* $\mathbb{T}$*-comodels in* $\mathcal{C}$ *and* ${}^{\mathbb{T}}U \colon {}^{\mathbb{T}}\mathcal{C} \to \mathcal{C}$ *for the forgetful functor.*

If we say simply "model" or "comodel", we will by default mean model or comodel in $\mathcal{S}\mathrm{et}$. As explained in [31, 32], set-based comodels provide deterministic environments suitable for evaluating computations with effects from $\mathbb{T}$. Before recalling how this works, we first unfold the notion of comodel (in $\mathcal{S}\mathrm{et}$) for our running examples, noting that when $\mathcal{C} = \mathcal{S}\mathrm{et}$, the copower $A \cdot X$ is simply the cartesian product $A \times X$.

Example 2.16 *A comodel* $\boldsymbol{S}$ *of the theory of V-valued input is a state machine which responds to requests for V-characters; it comprises a set of states S and a map* $[\![\mathsf{read}]\!]_{\boldsymbol{S}} = (g, n) \colon S \to V \times S$ *assigning to each* $s \in S$ *a character* $g(s) \in V$ *to be read and a new state* $n(s) \in S$.

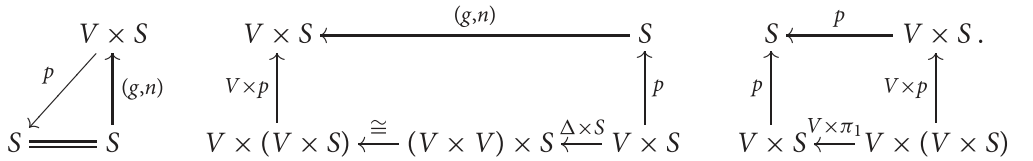
Example 2.17 *A comodel* $\boldsymbol{S}$ *of the theory of V-valued output is a state machine which changes its state in response to V-characters: it comprises a set of states S and maps* $[\![\mathsf{write}_v]\!]_{\boldsymbol{S}} \colon S \to S$ *for each* $v \in V$, *or equally, a single map* $p \colon V \times S \to S$, *providing for each character* $v \in V$ *and state* $s \in S$ *a new state* $p(v, s) \in S$.

Example 2.18 *A comodel* $\boldsymbol{S}$ *of the theory of V-valued read-only state comprises a set S together with a function* $[\![\mathsf{get}]\!]_{\boldsymbol{S}} = (g, n) \colon S \to V \times S$ *satisfying*

$$n(s) = s \qquad \text{and} \qquad \big(g(s), g(n(s)), n(n(s))\big) = \big(g(s), g(s), n(s)\big)\,. \tag{7}$$

The first axiom allows us to ignore n, and moreover implies the second axiom; whence a comodel amounts to nothing more than a set S and a function $g \colon S \to V$.

Example 2.19 *A comodel of the theory of V-valued state involves a set of states S together with maps* $(g, n) \colon S \to V \times S$ *and* $p \colon S \times V \to S$ *rendering commutative the diagrams*



These are equally the conditions that

$$p(g(s), n(s)) = s,\ \ n(p(v, s)) = p(v, s),\ \ g(p(v, s)) = v \ \text{ and } \ p(v', p(v, s)) = p(v', s)\,,$$

the first two of which imply $n = \mathrm{id}_S$. *Thus, to give the comodel is to give the set S together with maps* $g \colon S \to V$ *and* $p \colon V \times S \to S$ *satisfying the axioms*

$$p(g(s), s) = s \qquad g(p(v, s)) = v \quad \text{and} \quad p(v', p(v, s)) = p(v', s)\,;$$

as noted in [37], this is precisely a (very well-behaved) lens in the sense of [12].

A $\mathbb{T}$-comodel allows us to evaluate $\mathbb{T}$-computations by way of the derived operations of Definition 2.11. Indeed, given a comodel $\boldsymbol{S} \in {}^{\mathbb{T}}\mathcal{S}\mathrm{et}$ and a term $t \in T(A)$, the derived co-operation

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

$[\![t]\!] : S \to A \times S$ is defined by the recursive clauses

$$\begin{aligned} a \in A &\implies [\![a]\!](s) = (a, s) \\ \text{and } \sigma \in \Sigma, t \in T(A)^{|\sigma|} &\implies [\![\sigma(t)]\!](s) = [\![t_i]\!](s') \text{ where } [\![\sigma]\!](s) = (i, s') \,. \end{aligned} \tag{8}$$

(Here, and henceforth, we drop the subscript $\boldsymbol{S}$ where confusion seems unlikely.) The semantics of this is clear: given a $\mathbb{T}$-computation $\sigma(t) \in T(A)$ and starting state s, we respond to the request for a $|\sigma|$-element posed by the outermost operation symbol of $\sigma(t)$ by evaluating $[\![\sigma]\!](s)$ to obtain $i \in |\sigma|$ along with a new state s'; substituting this i into t yields the simpler computation t_i which we now run from state s'. When we hit a value $a \in A$ we return it along with the final state reached.

Example 2.20 *Consider the theory of $\mathbb{N}$-valued input, and the term from (2), which we may equally write as $t = \mathsf{read}(\lambda n.\, \mathsf{read}(\lambda m.\, n + m)) \in T(\mathbb{N})$. If $\boldsymbol{S}$ is the comodel with $S = \{s, s', s''\}$ and $[\![\mathsf{read}]\!] : S \to \mathbb{N} \times S$ given as to the left in:*

$$\begin{aligned} s &\mapsto (7, s') & \qquad s &\mapsto (18, s'') \\ s' &\mapsto (11, s'') & \qquad s' &\mapsto (24, s'') \\ s'' &\mapsto (13, s'') & \qquad s'' &\mapsto (26, s'') \,, \end{aligned}$$

then $[\![t]\!] : S \to \mathbb{N} \times S$ is given as to the right. For example, for $[\![t]\!](s)$ we calculate that $[\![\mathsf{read}(\lambda n.\, \mathsf{read}(\lambda m.\, n + m))]\!](s) = [\![\mathsf{read}(\lambda m.\, 7 + m)]\!](s') = [\![7 + 11]\!](s'') = (18, s'')$.

We conclude our discussion of models and comodels by describing the functoriality of the assignment $\mathbb{T} \mapsto \mathcal{C}^{\mathbb{T}}$.

Definition 2.21 (Semantics and cosemantics). *For a category $\mathcal{C}$ with powers, the* semantics functor $\mathcal{S}\mathrm{em}_{\mathcal{C}} : \mathcal{A}\mathrm{lg}\mathcal{T}\mathrm{h}^{\mathrm{op}} \to \mathcal{C}\mathrm{AT}/\mathcal{C}$ *is given by $\mathbb{T} \mapsto (U^{\mathbb{T}} : \mathcal{C}^{\mathbb{T}} \to \mathcal{C})$ on objects, while on maps, an interpretation $f : \mathbb{T}_1 \to \mathbb{T}_2$ is taken to the functor $f^* : \mathcal{C}^{\mathbb{T}_2} \to \mathcal{C}^{\mathbb{T}_1}$ over $\mathcal{C}$ acting via $(X, [\![-]\!]_X) \mapsto (X, [\![(-)^f]\!]_X)$.*

Dually, for any category $\mathcal{C}$ with copowers, we define the *cosemantics functor* $\mathcal{C}\mathrm{osem}_{\mathcal{C}} : \mathcal{A}\mathrm{lg}\mathcal{T}\mathrm{h}^{\mathrm{op}} \to \mathcal{C}\mathrm{AT}/\mathcal{C}$ by $\mathbb{T} \mapsto ({}^{\mathbb{T}}U : {}^{\mathbb{T}}\mathcal{C} \to \mathcal{C})$ on objects, and on morphisms in the same manner as above; more formally, we have $\mathcal{C}\mathrm{osem}_{\mathcal{C}} = \mathcal{S}\mathrm{em}_{\mathcal{C}^{\mathrm{op}}}(-)^{\mathrm{op}}$.

The functoriality of (co)semantics implies that theories $\mathbb{T}$ and $\mathbb{T}'$ which are isomorphic in $\mathcal{A}\mathrm{lg}\mathcal{T}\mathrm{h}$ have the same (co)models in any category with (co)powers $\mathcal{C}$; we call such theories *Morita equivalent*.

Example 2.22 *Let $h : V \to W$ be a function, and let $f : \mathbb{T}_1 \to \mathbb{T}_2$ be the interpretation of V-valued output into W-valued state of Example 2.8. For each comodel*

$$\boldsymbol{S} = (S, g : S \to W, p : S \times W \to S)$$

of W-valued state, the associated comodel $f^*\boldsymbol{S}$ of V-valued output is $(S, p \circ (1 \times h) : S \times V \to S)$.

Example 2.23 *Let $h : W \to V$ be a function, and let $f : \mathbb{T}_1 \to \mathbb{T}_2$ be the interpretation of V-valued read-only state into W-valued state of Example 2.9. For each comodel*

$$\boldsymbol{S} = (S, g : S \to W, p : S \times W \to S)$$

of W-valued state, the associated comodel $f^*\boldsymbol{S}$ of V-valued read-only state is $(S, hg : S \to V)$.

2.3. The associated monad

Finally in this section, we recall how an algebraic theory gives rise to a monad on $\mathcal{S}\mathrm{et}$, and the manner in which this interacts with semantics. We specify our monads as Kleisli triples in the style of [23, Exercise 1.3.12].

Definition 2.24 (Associated monad). *The* associated monad T *of an algebraic theory $\mathbb{T}$ has action on objects $A \mapsto T(A)$; unit maps $\eta_A : A \to T(A)$ given by inclusion of variables; and Kleisli*

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

$u^\dagger\colon T(A) \to T(B)$ *of* $u\colon A \to T(B)$ *given by* $t \mapsto t(u)$. *The assignment* $\mathbb{T} \mapsto \mathsf{T}$ *is the action on objects of the* associated monad functor $\mathrm{Ass}\colon \mathcal{A}\mathrm{lg}\mathcal{T}\mathrm{h} \to \mathcal{M}\mathrm{nd}(\mathcal{S}\mathrm{et})$, *which on morphisms takes an interpretation* $f\colon \mathbb{T}_1 \to \mathbb{T}_2$ *to the monad morphism* $\mathsf{T}_1 \to \mathsf{T}_2$ *whose components* $T_1(A) \to T_2(A)$ *are the assignments* $t \mapsto t^f$ *defined as in (4).*

Proposition 2.25 *The associated monad functor* $\mathcal{A}\mathrm{lg}\mathcal{T}\mathrm{h} \to \mathcal{M}\mathrm{nd}(\mathcal{S}\mathrm{et})$ *is full and faithful, and a monad* T *is in its essential image just when it is accessible.*

Here, a monad on $\mathcal{S}\mathrm{et}$ is *accessible* if its underlying endofunctor is accessible, in the sense of being a small colimit of representable functors. There are well-known monads on $\mathcal{S}\mathrm{et}$ which are not accessible, for example the power-set monad $\mathbb{P}$ and the continuation monad $V^{V^{(-)}}$; nonetheless, we may treat any monad T on $\mathcal{S}\mathrm{et}$ "as if it were induced by an algebraic theory" by adopting the following conventions: if $a \in A$, then we may write $a \in T(A)$ in place of $\eta_A(a) \in T(A)$, and if $t \in T(A)$ and $u \in T(B)^A$, then we may write $t(u)$ in place of $u^\dagger(t)$.

We now discuss how the model and comodel semantics of an algebraic theory can be expressed in terms of the associated monad.

Definition 2.26 (T-models and comodels) *Let* T *be a monad on* $\mathcal{S}\mathrm{et}$ *and let* $\mathcal{C}$ *be a category with powers. A* T*-model* $\boldsymbol{X}$ *in* $\mathcal{C}$ *is an object* $X \in \mathcal{C}$ *together with operations* $[\![t]\!]_X\colon X^A \to X$ *for every set* A *and* $t \in T(A)$, *subject to the axioms*

$$[\![a]\!]_X = \pi_a \qquad \text{and} \qquad [\![t(u)]\!]_X = X^B \xrightarrow{([\![u_a]\!]_X)_{a\in A}} X^A \xrightarrow{[\![t]\!]_X} X \tag{9}$$

for all $a \in A$ *and all* $t \in T(A)$ *and* $u \in T(B)^A$. *We write* $\mathcal{C}^\mathsf{T}$ *for the category of* T*-models in* $\mathcal{C}$, *whose maps* $\boldsymbol{X} \to \boldsymbol{Y}$ *are* $\mathcal{C}$*-maps* $f\colon X \to Y$ *with* $[\![t]\!]_Y \circ f^A = f \circ [\![t]\!]_X$ *for all sets* A *and all* $t \in T(A)$; *we write* $U^\mathsf{T}\colon \mathcal{C}^\mathsf{T} \to \mathcal{C}$ *for the forgetful functor.*

If $\mathcal{C}$ is a category with copowers then a T-comodel in $\mathcal{C}$ is a T-model in $\mathcal{C}^{\mathrm{op}}$, involving cooperations $[\![t]\!]_X\colon X \to A \cdot X$ subject to the dual axioms

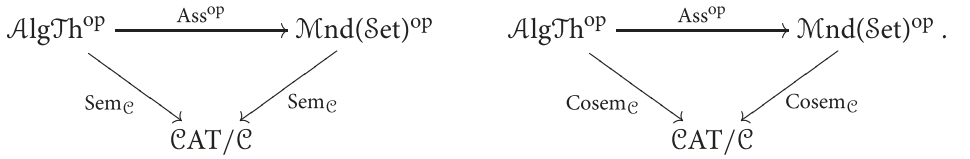
$$[\![a]\!]_X = \nu_a \qquad \text{and} \qquad [\![t(u)]\!]_X = X \xrightarrow{[\![t]\!]_X} A \cdot X \xrightarrow{\langle [\![u_a]\!]_X \rangle_{a\in A}} B \cdot X\,. \tag{10}$$

We write ${}^\mathsf{T}U\colon {}^\mathsf{T}\mathcal{C} \to \mathcal{C}$ for the forgetful functor from the category of T-comodels.

Definition 2.27 (Semantics and cosemantics). *For any category* $\mathcal{C}$ *with powers, the* semantics functor $\mathrm{Sem}_\mathcal{C}\colon \mathcal{M}\mathrm{nd}(\mathcal{S}\mathrm{et})^{\mathrm{op}} \to \mathcal{C}\mathrm{AT}/\mathcal{C}$ *is given by* $\mathsf{T} \mapsto (U^\mathsf{T}\colon \mathcal{C}^\mathsf{T} \to \mathcal{C})$ *on objects; while a monad morphism* $f\colon \mathsf{T}_1 \to \mathsf{T}_2$ *is taken to the functor* $f^*\colon \mathcal{C}^{\mathsf{T}_2} \to \mathcal{C}^{\mathsf{T}_1}$ *over* $\mathcal{C}$ *acting via* $(X, [\![-]\!]_X) \mapsto (X, [\![f(-)]\!]_X)$. *For a category* $\mathcal{C}$ *with copowers, we define the cosemantics functor by* $\mathrm{Cosem}_\mathcal{C} := \mathrm{Sem}_{\mathcal{C}^{\mathrm{op}}}(-)^{\mathrm{op}}\colon \mathcal{M}\mathrm{nd}(\mathcal{S}\mathrm{et})^{\mathrm{op}} \to \mathcal{C}\mathrm{AT}/\mathcal{C}$.

The following result, which again is entirely standard, tells us that we lose no semantic information in passing from an algebraic theory to the associated monad.

Proposition 2.28 *For any category* $\mathcal{C}$ *with powers (respectively, copowers), the triangle to the left (respectively, right) below commutes to within natural isomorphism:*



In light of this result, we will henceforth prefer to deal with accessible monads, though noting as we go along any simplifications afforded by having available a presentation via an algebraic theory.

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

3. Presheaf Monads and Comonads

3.1. Presheaf monads and comonads

In this section, we describe and study the presheaf monads and comonads which will be crucial to our main results. This is largely revision from the literature, though Propositions 3.8 and 3.12 are novel.

Definition 3.1 (Presheaf monad and comonad). *Let* $\mathbb{B}$ *be a small category. The* presheaf monad $\mathsf{T}_{\mathbb{B}}$ *and the* presheaf comonad $\mathsf{Q}_{\mathbb{B}}$ *are the accessible monad and accessible comonad on* $\mathcal{S}\mathrm{et}$ *induced by the respective adjunctions:*

$$\mathcal{S}\mathrm{et}^{\mathbb{B}^{\mathrm{op}}} \underset{\mathrm{res}_{J^{\mathrm{op}}}}{\overset{\mathrm{lan}_{J^{\mathrm{op}}}}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathcal{S}\mathrm{et}^{\mathrm{ob}(\mathbb{B})} \underset{\Pi}{\overset{\Delta}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathcal{S}\mathrm{et} \qquad \mathcal{S}\mathrm{et}^{\mathbb{B}} \underset{\mathrm{res}_J}{\overset{\mathrm{ran}_J}{\underset{\longrightarrow}{\overset{\longleftarrow}{\top}}}} \mathcal{S}\mathrm{et}^{\mathrm{ob}(\mathbb{B})} \underset{\Sigma}{\overset{\Delta}{\underset{\longrightarrow}{\overset{\longleftarrow}{\top}}}} \mathcal{S}\mathrm{et}\,, \tag{11}$$

where $J\colon \mathrm{ob}(\mathbb{B}) \to \mathbb{B}$ *is the inclusion of objects, and where res, lan and ran denote restriction, left Kan extension and right Kan extension. If we write* $\mathbb{B}_b$ *for the set of all* $\mathbb{B}$*-maps with domain b, then the underlying endofunctors are given by*

$$T_{\mathbb{B}}(A) = \textstyle\prod_{b\in\mathbb{B}} \mathbb{B}_b \times A \qquad \text{and} \qquad Q_{\mathbb{B}}(A) = \textstyle\sum_{b\in\mathbb{B}} A^{\mathbb{B}_b}\,;$$

the unit and multiplication for $\mathsf{T}_{\mathbb{B}}$ *are given by*

$$\begin{aligned} \eta_A &\colon A \to \textstyle\prod_b (\mathbb{B}_b \times A) & \mu_A &\colon \textstyle\prod_b (\mathbb{B}_b \times \prod_{b'} (\mathbb{B}_{b'} \times A)) \to \prod_b (\mathbb{B}_b \times A) \\ a &\mapsto \lambda b.\, (1_b, a) & \lambda b.\, (f_b, \lambda b'.\, (g_{bb'}, a_{bb'})) &\mapsto \lambda b.\, (g_{b,\mathrm{cod}(f_b)} \circ f_b, a_{b,\mathrm{cod}(f_b)})\,. \end{aligned}$$

while the counit and comultiplication for $\mathsf{Q}_{\mathbb{B}}$ *are given by*

$$\begin{aligned} \varepsilon_A &\colon \textstyle\sum_b A^{\mathbb{B}_b} \to A & \delta_A &\colon \textstyle\sum_b A^{\mathbb{B}_b} \to \sum_b \left(\sum_{b'} A^{\mathbb{B}_{b'}}\right)^{\mathbb{B}_b} \\ (b, \varphi) &\mapsto \varphi(1_b) & (b, \varphi) &\mapsto \left(b, \lambda f.\, (\mathrm{cod}(f), \lambda g.\, \varphi(gf))\right). \end{aligned} \tag{12}$$

We call a general monad T *on* $\mathcal{S}\mathrm{et}$ *a presheaf monad if it is isomorphic to some* $\mathsf{T}_{\mathbb{B}}$*, and correspondingly on the comonad side.*

Presheaf monads and comonads have been considered in computer science; in [8] they are termed "dependently typed update monads" and "dependently typed coupdate comonads", respectively, but both have a longer history, as we now recall.

To the comonad side, we note that the underlying endofunctor of a presheaf comonad is *polynomial*, i.e., a coproduct of representable functors. Such endofunctors are exactly those which arise as the interpretations of set-based *containers* [1], and in [5], this was enhanced to a characterisation of polynomial comonads as the interpretations of so-called *directed containers*. Now, as observed in [8], directed containers correspond bijectively to small categories, and so we conclude that the presheaf comonads on $\mathcal{S}\mathrm{et}$ are precisely the polynomial comonads. For self-containedness, we include a short direct proof of this fact.

Proposition 3.2 *For a comonad* Q *on* $\mathcal{S}\mathrm{et}$*, the following conditions are equivalent:*

(i) Q *is a presheaf comonad;*
(ii) The underlying endofunctor Q is a coproduct of representables;
(iii) The underlying endofunctor Q preserves connected limits.

Proof. Clearly (i) $\Rightarrow$ (ii), and (ii) $\Leftrightarrow$ (iii) is standard category theory due to Diers [10]; so it remains to show (ii) $\Rightarrow$ (i). Suppose, then, that $Q = \sum_{b\in B} (-)^{E_b}$ is a coproduct of representables. By the Yoneda lemma, giving $\varepsilon\colon Q \Rightarrow 1$ is equivalent to giving the elements $1_b := \varepsilon_{E_b}(b, \lambda f.\, f) \in E_b$ for each $b \in B$. Similarly, giving $\delta\colon Q \Rightarrow QQ$ is equivalent to giving for each $b \in B$ an element

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

of $QQ(E_b)$, i.e., elements $\alpha(b) \in B$ and $\lambda f.\,(c(f), \rho_f)\colon E_{\alpha(b)} \to \sum_{b'} E_b^{E_{b'}}$. Now the three comonad axioms correspond under the Yoneda lemma to the following assertions:

- The axiom $\varepsilon Q \circ \delta = 1_Q$ asserts that $\alpha(b) = b$ and $\rho_f(1_{c(f)}) = f$;
- The axiom $Q\varepsilon \circ \delta = 1_Q$ asserts that $c(1_b) = b$ and $\rho_{1_b} = \mathrm{id}_{E_b}$;
- The axiom $Q\delta \circ \delta = \delta Q \circ \delta$ asserts that $c(g) = c(\rho_f(g))$ and $\rho_f \circ \rho_g = \rho_{\rho_f(g)}$.

But these are precisely the data and axioms of a small category $\mathbb{B}$ with object-set B, with $\mathbb{B}_b = E_b$, with identities 1_b, with codomain map c, and with precomposition by f given by ρ_f; and on defining $\mathbb{B}$ in this way, we clearly have $\mathsf{Q} = \mathsf{Q}_{\mathbb{B}}$. □

To the monad side, presheaf monads seem to have been first considered in [17, Example 8.7], in terms of a presentation as an algebraic theory. We now recall this presentation, though framing it in terms of the applications of [7].

Notation 3.3 Let Σ be a signature, and $q \in \Sigma(A)$ and $t, u \in \Sigma(B)$ terms where without loss of generality A is disjoint from B. For any $i \in A$, we write $t \equiv_{q,i} u$ (read as "t and u are equal in the ith place of q") as an abbreviation for the equation

$$q\Big(\lambda a.\ \begin{cases} t(\vec{b}) & \text{if } a = i \\ a & \text{if } a \neq i \end{cases}\Big) \equiv q\Big(\lambda a.\ \begin{cases} u(\vec{b}) & \text{if } a = i \\ a & \text{if } a \neq i \end{cases}\Big) \qquad \text{in } \Sigma(A \cup B \setminus \{i\})\,.$$

Example 3.4 (Dependently typed update). *Let $\mathbb{B}$ be a small category, whose objects we view as* values, *and whose arrows $b \to b'$ we view as* updates *from b to b'. The theory of* $\mathbb{B}$-valued dependently typed update *is generated by an* $\mathrm{ob}(\mathbb{B})$*-ary operation* get *satifying the axioms of read-only state, together with a unary operation* upd_f *for each morphism $f\colon b \to b'$ in $\mathbb{B}$, subject to the equations*

$$\mathsf{upd}_f(x) \equiv_{\mathsf{get},c} x \qquad \text{for } f\colon b \to b' \text{ and } c \neq b \text{ in } \mathbb{B}; \tag{13}$$

$$\mathsf{upd}_f(\mathsf{get}(\lambda a.\, x_a)) \equiv_{\mathsf{get},b} \mathsf{upd}_f(x_{b'}) \qquad \text{for } f\colon b \to b' \text{ in } \mathbb{B}; \tag{14}$$

$$\mathsf{upd}_{1_b}(x) \equiv x \qquad \text{for } b \in \mathrm{ob}(\mathbb{B}); \tag{15}$$

$$\mathsf{upd}_f(\mathsf{upd}_g(x)) \equiv_{\mathsf{get},b} \mathsf{upd}_{gf}(x) \qquad \text{for } f\colon b \to b', g\colon b' \to b'' \text{ in } \mathbb{B}. \tag{16}$$

The intended semantics is that get *reads a value associated with the current state; while* upd_f*, for $f\colon b \to b'$ in $\mathbb{B}$, attempts to update the value b to b' via f. If the value of the current state is not b, then the update fails (the first axiom above); while if the value is b, then we move to a new state with associated value b' (the second axiom). The remaining axioms assert that updates compose as expected.*

We now justify our nomenclature by showing that the theory of $\mathbb{B}$-valued dependently typed update generates the presheaf monad $\mathsf{T}_{\mathbb{B}}$, which as we have explained, is equally an example of a dependently typed update monad as in [7].

Proposition 3.5 *For any small category $\mathbb{B}$, the theory of $\mathbb{B}$-valued dependently typed update generates the presheaf monad $\mathsf{T}_{\mathbb{B}}$.*

Proof. For each set A, we make $T_{\mathbb{B}}(A) = \prod_b (\mathbb{B}_b \times A)$ a model of the theory $\mathbb{T}$ of dependently-typed update by taking $[\![\mathsf{get}]\!]\big(\lambda b.\lambda c.\,(g_{bc}, a_{bc})\big) = \lambda b.\,(g_{bb}, a_{bb})$ and

$$[\![\mathsf{upd}_f]\!](\lambda c.\,(g_c, a_c)) = \lambda c.\ \begin{cases} (g_c, a_c) & \text{if } c \neq b; \\ (g_{b'} f, a_{b'}) & \text{if } c = b \end{cases} \qquad \text{for } f\colon b \to b' \text{ in } \mathbb{B}.$$

It is easy to verify that an equation $t \equiv_{\mathsf{get},b} u$ holds in $T_{\mathbb{B}}(A)$ precisely when the interpretations $[\![t]\!]$ and $[\![u]\!]$ have the same postcomposition with the projection $\pi_b\colon \prod_b (\mathbb{B}_b \times A) \to \mathbb{B}_b \times A$; whence the axioms for dependently typed update are satisfied. Thus, we may extend $\eta_A\colon A \to \prod_b (\mathbb{B}_b \times A)$ uniquely to a homomorphism $p_A\colon T(A) \to \prod_b (\mathbb{B}_b \times A)$; we also have

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

a function $i_A \colon \prod_b (\mathbb{B}_b \times A) \to T(A)$ given by $\lambda c.\,(g_c, a_c) \mapsto \mathsf{get}(\lambda c.\,\mathsf{upd}_{g_c}(a_c))$ which we claim is also a model homomorphism. It commutes with get since $\mathsf{get}(\lambda b.\,\mathsf{get}(\lambda c.\,\mathsf{upd}_{g_{bc}}(a_{bc}))) = \mathsf{get}(\lambda c.\,\mathsf{upd}_{g_{cc}}(a_{cc}))$. To see it commutes with upd_f for $f \colon b \to b'$, we observe that

$$\mathsf{upd}_f(\mathsf{get}(\lambda c.\,\mathsf{upd}_{g_c}(a_c)) \equiv_{\mathsf{get},b} \mathsf{upd}_f(\mathsf{upd}_{g_{b'}}(a_{b'})) \equiv_{\mathsf{get},b} \mathsf{upd}_{g_{b'}f}(a_{b'})$$
$$\text{and } \mathsf{upd}_f(\mathsf{get}(\lambda c.\,\mathsf{upd}_{g_c}(a_c)) \equiv_{\mathsf{get},c} \mathsf{get}(\lambda c.\,\mathsf{upd}_{g_c}(a_c)) \equiv_{\mathsf{get},c} \mathsf{upd}_{g_c}(a_c) \text{ for } c \neq b,$$

from which it follows that

$$\mathsf{upd}_f(\mathsf{get}(\lambda c.\,\mathsf{upd}_{g_c}(a_c)) = \mathsf{get}\Big(\lambda c.\,\begin{cases}\mathsf{upd}_{g_c}(a_c) & \text{if } c \neq b;\\ \mathsf{upd}_{g_{b'}f}(a_{b'}) & \text{if } c = b\end{cases}\Big)$$

as desired. Since $\mathsf{get}(\lambda b.\,\mathsf{upd}_{1_b}(a)) \equiv \mathsf{upd}_{1_b}(a) \equiv a$, we have $i_A(\eta_A(a)) = a$, from which it follows that $i_A p_A = 1_{T(A)}$. On the other hand, $p_A i_A = 1$ by a short calculation, and so p_A and i_A are mutually inverse. In this way, we obtain a natural isomorphism $T \cong T_{\mathbb{B}}$, which by construction is compatible with the units of the monads T and $\mathsf{T}_{\mathbb{B}}$. Compatibility with the multiplications follows since:

$$\begin{aligned}\mathsf{get}(\lambda b.\,\mathsf{upd}_{f_b}(\mathsf{get}(\lambda c.\,\mathsf{upd}_{g_{bc}}(a_{bc})))) &\equiv_{\mathbb{T}} \mathsf{get}(\lambda b.\,\mathsf{upd}_{f_b}(\mathsf{upd}_{g_{b,\mathsf{c}(f_b)}}(a_{b,\mathsf{c}(f_b)})))\\ &\equiv_{\mathbb{T}} \mathsf{get}(\lambda b.\,\mathsf{upd}_{g_{b,\mathsf{c}(f_b)} \circ f_b}(a_{b,\mathsf{c}(f_b)}))\,.\end{aligned}$$

□

3.2. *Morphisms of presheaf monads and comonads*

We now examine morphisms between presheaf monads and comonads. Beginning again on the comonad side, we observe, as in [8], that comonad morphisms between presheaf comonads do not correspond to functors, but rather to *cofunctors*:

Definition 3.6 (Cofunctor). *[3, 16] A cofunctor $F \colon \mathbb{B} \rightsquigarrow \mathbb{C}$ between small categories comprises an action on objects $F \colon \mathrm{ob}(\mathbb{B}) \to \mathrm{ob}(\mathbb{C})$ together with actions on morphisms $F_b \colon \mathbb{C}_{Fb} \to \mathbb{B}_b$ for each $b \in \mathbb{B}$, subject to the axioms:*

(i) $F(\mathrm{cod}(F_b(f))) = \mathrm{cod}(f)$ *for all* $f \in \mathbb{C}_{Fb}$;
(ii) $F_b(1_{Fb}) = 1_b$ *for all* $b \in \mathbb{B}$;
(iii) $F_b(gf) = F_{\mathrm{cod}(F_b f)}(g) \circ F_b(f)$ *for all* $f \in \mathbb{C}_{Fb}$ *and* $g \in \mathbb{C}_{\mathrm{cod}(f)}$.

We write $\mathcal{C}\mathrm{of}$ *for the category of small categories and cofunctors.*

In what follows, $\mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et})$ denotes the category of *accessible* comonads on $\mathcal{S}\mathrm{et}$.

Proposition 3.7 *Taking presheaf comonads is the action on objects of a fully faithful functor $\mathsf{Q}_{(-)} \colon \mathcal{C}\mathrm{of} \to \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et})$, which on morphisms sends a cofunctor $F \colon \mathbb{B} \rightsquigarrow \mathbb{C}$ to the comonad morphism $\mathsf{Q}_F \colon \mathsf{Q}_{\mathbb{B}} \to \mathsf{Q}_{\mathbb{C}}$ with components*

$$\begin{aligned}\textstyle\sum_{b \in \mathbb{B}} A^{\mathbb{B}_b} &\to \textstyle\sum_{c \in \mathbb{C}} A^{\mathbb{C}_c}\\ (b, \varphi) &\mapsto (Fb, \varphi \circ F_b)\,.\end{aligned} \tag{17}$$

This result is again due to [8], but we sketch a proof for self-containedness.

Proof. Let $\mathbb{B}$ and $\mathbb{C}$ be small categories. As $Q_{\mathbb{B}} = \sum_{b \in B} (-)^{\mathbb{B}_b}$, we see once again by the Yoneda lemma that giving a natural transformation $\alpha \colon Q_{\mathbb{B}} \Rightarrow Q_{\mathbb{C}}$ is equivalent to giving elements $\alpha_{\mathbb{B}_b}(b, 1_b) \in Q_{\mathbb{C}}(\mathbb{B}_b)$; and if we write these elements as pairs $(Fb \in \mathbb{C}, F_b \colon \mathbb{C}_{Fb} \to \mathbb{B}_b)$, then α itself must have components given as in (17). Similar arguments to the proof of Proposition 3.2

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

now show that α commutes with the comonad counits and comultiplication precisely when the assignments $b \mapsto (Fb, F_b)$ satisfy the axioms (i)–(iii) for a cofunctor. □

On the monad side, it turns out that monad morphisms between presheaf monads are also cofunctors between the corresponding categories. This is not quite as straightforward to see, and for the moment we record only the weaker statement that cofunctors induce morphisms of presheaf monads; the full claim will be proved in Proposition 7.8 below.

Once again, in what follows, $\mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})$ will denote the category of *accessible* monads on $\mathcal{S}\mathrm{et}$.

Proposition 3.8 *Taking presheaf monads is the action on objects of a functor* $\mathsf{T}_{(-)} \colon \mathcal{C}\mathrm{of} \to \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}}$, *which on morphisms sends a cofunctor* $F \colon \mathbb{B} \rightsquigarrow \mathbb{C}$ *to the monad morphism* $\mathsf{T}_F \colon \mathsf{T}_{\mathbb{C}} \to \mathsf{T}_{\mathbb{B}}$ *with components*

$$\begin{aligned} \textstyle\prod_{c \in \mathbb{C}} (\mathbb{C}_c \times A) &\to \textstyle\prod_{b \in \mathbb{B}} (\mathbb{B}_b \times A) \\ \lambda c.\, (f_c, a_c) &\mapsto \lambda b.\, (F_b(f_{Fb}), a_{Fb})\,. \end{aligned} \tag{18}$$

Proof. It is easy to check that the components (18) are natural in A and compatible with the units and multiplications of the presheaf monads $\mathsf{T}_{\mathbb{B}}$ and $\mathsf{T}_{\mathbb{C}}$. □

3.3. Semantics

We now consider the semantics associated with presheaf comonads and monads. Starting again on the comonad side, it turns out that the adjunction (11) inducing the presheaf comonad $\mathsf{Q}_{\mathbb{B}}$ is comonadic, but not *strictly* so; thus, $\mathsf{Q}_{\mathbb{B}}$-coalgebras are not exactly presheaves $\mathbb{B} \to \mathcal{S}\mathrm{et}$, but only something equivalent:

Definition 3.9 (Left $\mathbb{B}$-set). *Let* $\mathbb{B}$ *be a small category. A left* $\mathbb{B}$-set *is a set* X *endowed with a projection map* $p \colon X \to \mathrm{ob}(\mathbb{B})$ *and an action* $* \colon \sum_{x \in X} \mathbb{B}_{p(x)} \to X$, *notated as* $(x, f) \mapsto f * x$, *satisfying the typing axiom* $p(f * x) = \mathrm{cod}(f)$ *and the functoriality axioms* $\mathrm{id} * x = x$ *and* $g * (f * x) = (g \circ f) * x$. *We write* $\mathbb{B}$-$\mathcal{S}\mathrm{et}$ *for the category of left* $\mathbb{B}$*-sets, whose morphisms are functions commuting with projections and actions. We write* $U^{\mathbb{B}} \colon \mathbb{B}\text{-}\mathcal{S}\mathrm{et} \to \mathcal{S}\mathrm{et}$ *for the forgetful functor* $(X, p, *) \mapsto X$.

Given a presheaf $X \colon \mathbb{B} \to \mathcal{S}\mathrm{et}$, we have a corresponding left $\mathbb{B}$-set given by $Y = \sum_{b \in \mathbb{B}} X(b)$ with the action map $(f \colon b \to c) * (b, x) = (c, X(f)(x))$; conversely, given a left $\mathbb{B}$-set $(Y, p, *)$, we have a corresponding presheaf $X \colon \mathbb{B} \to \mathcal{S}\mathrm{et}$ with $X(b) = p^{-1}(b)$ and $X(f)(x) = f * x$. These assignments give an equivalence of categories fitting into a commuting triangle

$$\begin{array}{ccc} [\mathbb{B}, \mathcal{S}\mathrm{et}] & \xrightarrow{\ \simeq\ } & \mathbb{B}\text{-}\mathcal{S}\mathrm{et} \\ & {\scriptstyle U^{\mathbb{B}}} \searrow \quad \swarrow {\scriptstyle V} & \\ & \mathcal{S}\mathrm{et} & \end{array} \tag{19}$$

where here V sends $X \colon \mathbb{B} \to \mathcal{S}\mathrm{et}$ to $\Sigma_{b \in B} X(b)$.

In [6], what we call a left $\mathbb{B}$-set was termed a *coalgebraic update lens*; the following result, which is immediate from the definitions, was also observed there.

Proposition 3.10 *For any small category* $\mathbb{B}$, *the category of Eilenberg–Moore* $\mathsf{Q}_{\mathbb{B}}$*-coalgebras is isomorphic to* $\mathbb{B}$-$\mathcal{S}\mathrm{et}$ *via an isomorphism commuting with the functors to* $\mathcal{S}\mathrm{et}$.

It is moreover easy to see that the isomorphisms of this proposition are natural with respect to cofunctors $F \colon \mathbb{B} \rightsquigarrow \mathbb{C}$. To make this precise, we require:

Definition 3.11 (Sum along a cofunctor). *Given a cofunctor* $F \colon \mathbb{B} \rightsquigarrow \mathbb{C}$ *between small categories, we define the functor* $\Sigma_F \colon \mathbb{B}\text{-}\mathcal{S}\mathrm{et} \to \mathbb{C}\text{-}\mathcal{S}\mathrm{et}$ *to act by*

$$(X \xrightarrow{p} \mathrm{ob}(\mathbb{B}),\ \textstyle\sum_{x \in X} \mathbb{B}_{p(x)} \xrightarrow{*} X) \ \mapsto\ (X \xrightarrow{Fp} \mathrm{ob}(\mathbb{C}), \textstyle\sum_{x \in X} \mathbb{C}_{F(p(x))} \xrightarrow{*^F} X)$$

where we write $f *^F x := F_{p(x)}(f) * x$.

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

The naturality of the isomorphisms in Proposition 3.10 in question is now expressed by commutativity of each square

$$\begin{array}{ccc} \mathcal{C}\mathrm{oalg}(\mathsf{Q}_{\mathbb{B}}) & \xrightarrow{\cong} & \mathbb{B}\text{-}\mathcal{S}\mathrm{et} \\ {\scriptstyle \mathcal{C}\mathrm{oalg}(\mathsf{Q}_F)}\downarrow & & \downarrow{\scriptstyle \Sigma_F} \\ \mathcal{C}\mathrm{oalg}(\mathsf{Q}_{\mathbb{C}}) & \xrightarrow[\cong]{} & \mathbb{C}\text{-}\mathcal{S}\mathrm{et}\,. \end{array}$$

Turning now to the presheaf monad $\mathsf{T}_{\mathbb{B}}$, it follows from the results of [17] that the category of $\mathsf{T}_{\mathbb{B}}$-models is equivalent to the category of presheaves $X\colon \mathbb{B}^{\mathrm{op}} \to \mathcal{S}\mathrm{et}$ which *either* have each $X(b)$ empty, *or* each $X(b)$ non-empty. However, we will be less interested in characterising the $\mathsf{T}_{\mathbb{B}}$-models in $\mathcal{S}\mathrm{et}$ than the $\mathsf{T}_{\mathbb{B}}$-*comodels*. We may exploit the fact that $\mathsf{T}_{\mathbb{B}}$ is generated by the theory of $\mathbb{B}$-valued dependently typed update to obtain such a characterisation.

Proposition 3.12 *For any small category $\mathbb{B}$, the category of comodels of the theory $\mathbb{T}_{\mathbb{B}}$ of $\mathbb{B}$-valued dependently-typed update is isomorphic to $\mathbb{B}$-$\mathcal{S}\mathrm{et}$ via a functor*

$$\begin{array}{ccc} \mathbb{B}\text{-}\mathcal{S}\mathrm{et} & \longrightarrow & \mathbb{T}_{\mathbb{B}}\,\mathcal{S}\mathrm{et} \\ & {\scriptstyle U^{\mathbb{B}}}\searrow \quad \swarrow{\scriptstyle \mathbb{T}_{\mathbb{B}}\,U} & \\ & \mathcal{S}\mathrm{et} & \end{array} \tag{20}$$

which sends a left $\mathbb{B}$-set $(X, p, *)$ to the $\mathbb{T}_{\mathbb{B}}$-comodel $\boldsymbol{X} = (X, [\![-]\!]_X)$ with

$$[\![\mathsf{get}]\!]_X(x) = (p(x), x) \qquad [\![\mathsf{upd}_f]\!]_X(x) = \begin{cases} x & \text{if } p(x) \neq \mathrm{dom}(f); \\ f * x & \text{if } p(x) = \mathrm{dom}(f). \end{cases}$$

Proof. A comodel of $\mathbb{B}$-valued dependently typed state is firstly, a comodel of $\mathrm{ob}(\mathbb{B})$-valued read-only state, i.e., a set X endowed with a function $p\colon X \to \mathrm{ob}(\mathbb{B})$. On top of this, we have functions $[\![\mathsf{upd}_f]\!]\colon X \to X$ for each $f\colon b \to b'$ in $\mathbb{B}$ which satisfy the equations (13)–(16). The first forces $[\![\mathsf{upd}_f]\!]$ to act trivially on the fibre $p^{-1}(c)$ for all $c \neq b$, while the second forces it to map $p^{-1}(b)$ into $p^{-1}(b')$. So to give the $[\![\mathsf{upd}_f]\!]$'s satisfying (13) and (14) is equally to give functions $f * (-)\colon p^{-1}(b) \to p^{-1}(b')$ for each $f\colon b \to b'$ in $\mathbb{B}$. Now the last two axioms (15) and (16) impose the functoriality constraints $1_b * x = x$ and $g * (f * x) = (g \circ f) * x$, so that, in sum, a comodel of $\mathbb{B}$-valued dependently typed update can be identified with a left $\mathbb{B}$-set, via the identification given in the statement of the result. □

4. The Costructure–Cosemantics Adjunction

In this section, we construct the adjunction which is the main object of study of this paper. We begin by explaining how taking comodels yields a *cosemantics* functor from accessible monads to accessible comonads on $\mathcal{S}\mathrm{et}$. We then show that this functor has a left adjoint, as displayed below, which we term the *costructure* functor; and finally, we explain how this relates to the material of [19].

$$\mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}} \underset{\mathrm{Cosem}}{\overset{\mathrm{Costr}}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et}) \tag{21}$$

4.1. The cosemantics comonad of an accessible monad

Our first task is to show that the cosemantics functor of Example 2.27 yields the right adjoint functor in (21). We begin with the basic facts about Eilenberg–Moore semantics for comonads.

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

Definition 4.1 (Eilenberg–Moore semantics). *Let* $\mathcal{C}\mathrm{md}(\mathcal{C})$ *be the category of comonads in* $\mathcal{C}$. *The* Eilenberg–Moore semantics functor $\mathrm{EM}\colon \mathcal{C}\mathrm{md}(\mathcal{C}) \to \mathcal{C}\mathrm{AT}/\mathcal{C}$ *sends a comonad* $\mathsf{Q} = (Q, \varepsilon, \delta)$ *to the forgetful functor* $U^{\mathsf{Q}}\colon \mathcal{C}\mathrm{oalg}(\mathsf{Q}) \to \mathcal{C}$ *from its category of Eilenberg–Moore coalgebras, and sends* $f\colon \mathsf{Q} \to \mathsf{P}$ *to the functor* $\mathcal{C}\mathrm{oalg}(\mathsf{Q}) \to \mathcal{C}\mathrm{oalg}(\mathsf{P})$ *over* $\mathcal{C}$ *which acts on objects by* $(X, x\colon X \to QX) \mapsto (X, f_X \circ x\colon X \to PX)$.

Lemma 4.2 *For any category* $\mathcal{C}$, *the semantics functor* $\mathrm{EM}\colon \mathcal{C}\mathrm{md}(\mathcal{C}) \to \mathcal{C}\mathrm{AT}/\mathcal{C}$ *is full and faithful, and its essential image comprises the strictly comonadic functors.*

Proof. The first part is [9, Theorem 6.3]; the second is easy from the definitions. □

Here, a functor $V\colon \mathcal{D} \to \mathcal{C}$ is strictly comonadic if it has a right adjoint G, and the canonical comparison functor from $\mathcal{D}$ to the category of coalgebras for the comonad VG is an isomorphism. Concrete conditions for a functor to be strictly comonadic are given by the Beck comonadicity theorem [9, Theorem 3.14].

Proposition 4.3 *For an accessible* $\mathcal{S}\mathrm{et}$*-monad* T, *the forgetful functor* ${}^{\mathsf{T}}U\colon {}^{\mathsf{T}}\mathcal{S}\mathrm{et} \to \mathcal{S}\mathrm{et}$ *from the category of comodels is strictly comonadic for an accessible comonad.*

In the finitary case, this is [32, Theorem 2.2]; this more general form can be proven as a routine application of the theory of locally presentable categories. We omit this, as Theorem 5.16 provides an independent elementary argument.

Corollary 4.4 *The cosemantics functor* $\mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}} \to \mathcal{C}\mathrm{AT}/\mathcal{S}\mathrm{et}$ *factors as*

$$\mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}} \xrightarrow{\text{Cosem}} \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et}) \xrightarrow{\text{EM}} \mathcal{C}\mathrm{AT}/\mathcal{S}\mathrm{et}\,.$$

4.2. The costructure monad of an accessible comonad

We now show that the cosemantics functor $\mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}} \to \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et})$ has a left adjoint. This will arise from the "structure–semantics adjointness" of [11, 22], which we now recall.

Definition 4.5 (Endomorphism monad). *Let* $\mathcal{C}$ *be a category with powers which is not necessarily locally small. We say that* $X \in \mathcal{C}$ *is* tractable *if, for any set* A, *the collection of morphisms* $X^A \to X$ *form a set. For such an* X, *the* endomorphism *monad* $\mathsf{End}_{\mathcal{C}}(X)$ *on* $\mathcal{S}\mathrm{et}$ *has action on objects given by* $A \mapsto \mathcal{C}(X^A, X)$; *unit functions* $A \to \mathcal{C}(X^A, X)$ *given by* $a \mapsto \pi_a$; *and Kleisli extension* $u^\dagger\colon \mathcal{C}(X^A, X) \to \mathcal{C}(X^B, X)$ *of* $u\colon A \to \mathcal{C}(X^B, X)$ *given by* $t \mapsto t \circ (u_a)_{a \in A}$.

Note that endomorphism monads need *not* be accessible; for example, the endomorphism monad of $V \in \mathcal{S}\mathrm{et}$ is the non-accessible continuation monad $V^{V^{(-)}}$.

Lemma 4.6 *Let* $\mathcal{C}$ *be a category with powers, not necessarily locally small, and let* $X \in \mathcal{C}$ *be tractable. There is a bijection, natural in* T, *between monad morphisms* $\mathsf{T} \to \mathsf{End}_{\mathcal{C}}(X)$ *and* T*-model structures on* X.

Proof. To give a monad map $\mathsf{T} \to \mathsf{End}_{\mathcal{C}}(X)$ is to give functions $T(A) \to \mathcal{C}(X^A, X)$ for each set A, compatibly with units and Kleisli extensions. If we write the action of these functions as $t \mapsto [\![t]\!]_X$, then these compatibilities are precisely the conditions (5) to make the $[\![t]\!]_X$'s into a T-model structure on X. □

In the following result, we call a functor $V\colon \mathcal{A} \to \mathcal{C}$ *tractable* if it is tractable as an object of the (not necessarily locally small) functor category $[\mathcal{A}, \mathcal{C}]$.

Proposition 4.7 (Structure/semantics). *Let* $\mathcal{C}$ *be a category with powers. The semantics functor* $\mathrm{Sem}_{\mathcal{C}}\colon \mathcal{M}\mathrm{nd}(\mathcal{S}\mathrm{et})^{\mathrm{op}} \to \mathcal{C}\mathrm{AT}/\mathcal{C}$ *of Example 2.27 has a partial left adjoint at each tractable* $V\colon \mathcal{A} \to \mathcal{C}$, *given by the endomorphism monad* $\mathsf{End}_{[\mathcal{A},\mathcal{C}]}(V)$.

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

Proof. Let $V : \mathcal{A} \to \mathcal{C}$ be tractable, so that $\mathsf{End}_{[\mathcal{A},\mathcal{C}]}(V)$ exists. By Lemma 4.6, there is a bijection, natural in T, between monad morphisms $\mathsf{T} \to \mathsf{End}_{[\mathcal{A},\mathcal{C}]}(V)$ and T-model structures on V in $[\mathcal{A}, \mathcal{C}]$. Now since powers in $[\mathcal{A}, \mathcal{C}]$ are computed componentwise, T-model structures on V correspond, naturally in T, with liftings

$$\begin{array}{ccc} & & \mathcal{C}^{\mathsf{T}} \\ & \nearrow & \downarrow{\scriptstyle U^{\mathsf{T}}} \\ \mathcal{A} & \xrightarrow{V} & \mathcal{C} \end{array}$$

of V through U^{T}, i.e., with morphisms $V \to \mathrm{Sem}_{\mathcal{C}}(\mathsf{T})$ in $\mathcal{C}\mathrm{AT}/\mathcal{C}$. □

Because we are interested in comodels rather than models, we will apply this result in its dual form: thus, we speak of the *cotractability* of a functor $V : \mathcal{A} \to \mathcal{C}$, meaning that each collection $[\mathcal{A}, \mathcal{C}](V, A \cdot V)$ is a set, and the *coendomorphism* monad $\mathsf{Coend}_{[\mathcal{A},\mathcal{C}]}(V)$ with action on objects $A \mapsto [\mathcal{A}, \mathcal{C}](V, A \cdot V)$, providing the value at V of a partial left adjoint to $\mathrm{Cosem}_{\mathcal{C}} : \mathcal{M}\mathrm{nd}(\mathcal{S}\mathrm{et})^{\mathrm{op}} \to \mathcal{C}\mathrm{AT}/\mathcal{C}$.

Lemma 4.8 *Let* Q *be an accessible comonad on* $\mathcal{S}$et. *The forgetful functor from the category of Eilenberg–Moore coalgebras* $U^{\mathsf{Q}} : \mathcal{C}\mathrm{oalg}(\mathsf{Q}) \to \mathcal{S}\mathrm{et}$ *is cotractable, and the coendomorphism monad* $\mathsf{Coend}_{[\mathcal{C}\mathrm{oalg}(\mathsf{Q}),\mathcal{S}\mathrm{et}]}(U^{\mathsf{Q}})$ *is accessible.*

Proof. For cotractability, we show that for each set A, the collection of natural transformations $U^{\mathsf{Q}} \Rightarrow A \cdot U^{\mathsf{Q}}$ form a set. If we write G^{Q} for the right adjoint of U^{Q}, then transposing under the adjunction $(-) \circ G^{\mathsf{Q}} \dashv (-) \circ U^{\mathsf{Q}}$ yields

$$[\mathsf{Q}\text{-}\mathcal{C}\mathrm{oalg}, \mathcal{S}\mathrm{et}](U^{\mathsf{Q}}, A \cdot U^{\mathsf{Q}}) \cong [\mathcal{S}\mathrm{et}, \mathcal{S}\mathrm{et}](Q, A \cdot \mathrm{id}) , \tag{22}$$

whose right-hand side is a set since Q is accessible; whence also the left-hand side.

So $\mathsf{Coend}(U^{\mathsf{Q}})$ exists; to show accessibility, note that a natural transformation $Q \Rightarrow A \cdot \mathrm{id}$ is equally a pair of natural transformations $Q \Rightarrow \mathrm{id}$ and $Q \Rightarrow \Delta A$, where ΔA is the functor constant at A; and since $\mathcal{S}$et has a terminal object, to give $Q \Rightarrow \Delta A$ is equally to give a function $Q1 \to A$. We conclude that $\mathrm{Coend}(U^{\mathsf{Q}}) \cong (-)^{Q1} \times [\mathcal{S}\mathrm{et}, \mathcal{S}\mathrm{et}](Q, \mathrm{id})$, which is a small coproduct of representable functors, and hence accessible. □

Proposition 4.9 *The functor* $\mathrm{Cosem} : \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}} \to \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et})$ *of Corollary 4.4 admits a left adjoint* $\mathrm{Costr} : \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et}) \to \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}}$, *whose value at the accessible comonad* Q *is given by the coendomorphism monad* $\mathsf{Coend}(U^{\mathsf{Q}})$.

Proof. The preceding result shows that $\mathsf{Coend}(U^{\mathsf{Q}})$ exists and is accessible for each accessible comonad Q. Now by Lemma 4.2, Proposition 4.3 and the dual of Proposition 4.7, we have natural isomorphisms

$$\mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et})(\mathsf{Q}, \mathrm{Cosem}(\mathsf{T})) \cong \mathcal{C}\mathrm{AT}/\mathcal{S}\mathrm{et}(U^{\mathsf{Q}}, {}^{\mathsf{T}}U) \cong \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})(\mathsf{T}, \mathsf{Coend}(U^{\mathsf{Q}})) .$$ □

Remark 4.10 *For future use, we record the concrete form of the adjointness isomorphisms of the costructure–cosemantics adjunction. Given a comonad morphism* $\alpha : \mathsf{Q} \to \mathrm{Cosem}(\mathsf{T})$, *corresponding by Lemma 4.2 and Proposition 4.3 to a functor* H *as in:*

$$\begin{array}{ccc} \mathcal{C}\mathrm{oalg}(\mathsf{Q}) & \xrightarrow{H} & {}^{\mathsf{T}}\mathcal{S}\mathrm{et} , \\ & {\scriptstyle U^{\mathsf{Q}}} \searrow \quad \swarrow {\scriptstyle {}^{\mathsf{T}}U} & \\ & \mathcal{S}\mathrm{et} & \end{array}$$

the adjoint transpose $\bar{\alpha} : \mathsf{T} \to \mathsf{Coend}(U^{\mathsf{Q}})$ *of* α *sends* $t \in T(A)$ *to* $\bar{\alpha}(t) : U^{\mathsf{Q}} \Rightarrow A \cdot U^{\mathsf{Q}}$ *with components* $\bar{\alpha}(t)_{(X,x)} = [\![t]\!]_{H(X,x)} : X \to A \times X$.

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

4.3. Relation to duals and Sweedler duals

This completes our construction of the costructure–cosemantics adjunction (21); and in the rest of this section, we explain its relation to the notions of [19]. The main objects of study in *op. cit.* are the *interaction laws* between a monad T and a comonad Q on a category with products; these are natural families of morphisms $TX \times QY \to X \times Y$ which are compatible with the monad and comonad structures. In Section 3.4 of [19], the authors show that such monad–comonad interaction laws can also be expressed in terms of:

- Monad morphisms $\mathsf{T} \to \mathsf{Q}^\circ$, where Q° is the *dual monad* of Q;
- Comonad morphisms $\mathsf{Q} \to \mathsf{T}^\bullet$, where $\mathsf{T}^\bullet$ is the *Sweedler dual comonad* of T.

It may or may not be the case that the dual monad of a comonad, or the Sweedler dual comonad of a monad, exist; however, they do always exist when we are dealing with accessible monads and comonads on $\mathcal{S}\mathrm{et}$, and the definitions are as follows:

Definition 4.11 (Dual monad) *The* dual *of an accessible comonad* Q *on* $\mathcal{S}\mathrm{et}$ *is the accessible monad* Q° *with* $Q^\circ(A) = [\mathcal{S}\mathrm{et}, \mathcal{S}\mathrm{et}](Q, A \cdot \mathrm{id})$*, with unit map* $\eta_A \colon A \to Q^\circ A$ *given by*

$$a \qquad \mapsto \qquad Q \xrightarrow{\varepsilon} \mathrm{id} \xrightarrow{\nu_a} A \cdot \mathrm{id}$$

and with the Kleisli extension $u^\dagger \colon [\mathcal{S}\mathrm{et}, \mathcal{S}\mathrm{et}](Q, A \cdot \mathrm{id}) \to [\mathcal{S}\mathrm{et}, \mathcal{S}\mathrm{et}](Q, B \cdot \mathrm{id})$ *of the function* $u \colon A \to [\mathcal{S}\mathrm{et}, \mathcal{S}\mathrm{et}](Q, B \cdot \mathrm{id})$ *given by*

$$Q \xrightarrow{\tau} A \cdot \mathrm{id} \qquad \mapsto \qquad Q \xrightarrow{\delta} QQ \xrightarrow{\tau Q} A \cdot Q \xrightarrow{\langle u_a \rangle_{a \in A}} B \cdot \mathrm{id}\,.$$

The assignment $\mathsf{Q} \mapsto \mathsf{Q}^\circ$ *is the action on objects of the* dual monad *functor* $\mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et}) \to \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}}$*, whose action on morphisms takes a comonad map* $f \colon \mathsf{Q} \to \mathsf{P}$ *to the monad map* $\mathsf{P}^\circ \to \mathsf{Q}^\circ$ *with components* $\alpha \mapsto \alpha f$*.*

Definition 4.12 (Sweedler dual comonad) *The Sweedler dual of an accessible monad* T *on* $\mathcal{S}\mathrm{et}$ *is the accessible comonad* $\mathsf{T}^\bullet$ *providing the value at* T *of a right adjoint to the dual monad functor.*

We now show that, in fact, these constructions relating accessible monads and comonads are precisely the two directions of our adjunction (21).

Proposition 4.13 *For each* $\mathsf{Q} \in \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et})$*, there is a monad isomorphism* $\mathsf{Q}^\circ \cong \mathrm{Costr}(\mathsf{Q})$ *taking* $\alpha \colon Q \Rightarrow A \cdot \mathrm{id}$ *in* $Q^\circ A$ *to* $\tilde{\alpha} \colon U^{\mathsf{Q}} \Rightarrow A \cdot U^{\mathsf{Q}}$ *in* $\mathsf{Coend}(U^{\mathsf{Q}})(A)$ *with components*

$$\tilde{\alpha}_{(X,x)} = X \xrightarrow{x} QX \xrightarrow{\alpha_X} A \times X\,.$$

It follows that $\mathrm{Costr} \cong (-)^\circ \colon \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et}) \to \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}}$ *and, consequently, that* $\mathrm{Cosem} \cong (-)^\bullet \colon \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}} \to \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et})$*.*

Proof. Consider the category $\mathcal{X}$ whose objects are endofunctors of $\mathcal{S}\mathrm{et}$, and whose morphisms $F \to F'$ are natural transformations $FU^{\mathsf{Q}} \Rightarrow F'U^{\mathsf{Q}} \colon \mathcal{C}\mathrm{oalg}(\mathsf{Q}) \to \mathcal{S}\mathrm{et}$. It is easy to see that $\mathsf{Coend}(U^{\mathsf{Q}})$ is equally the coendomorphism monad of the object $\mathrm{id}_{\mathcal{S}\mathrm{et}} \in \mathcal{X}$. By transposing under the adjunction $(-) \circ G^{\mathsf{Q}} \dashv (-) \circ U^{\mathsf{Q}}$, we see that $\mathcal{X}$ is isomorphic to the co-Kleisli category $\mathcal{X}'$ of the comonad $(-) \circ \mathsf{Q}$ on $[\mathcal{S}\mathrm{et}, \mathcal{S}\mathrm{et}]$, and the coendomorphism monad of $\mathrm{id}_{\mathcal{S}\mathrm{et}}$ in $\mathcal{X}'$ is easily seen to be Q°. Thus, $\mathsf{Q}^\circ \cong \mathsf{Coend}(U^{\mathsf{Q}})$, and tracing through the correspondences shows this isomorphism to be given as in the statement of the result. □

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

5. Calculating the Cosemantics Functor

5.1. Cosemantics is valued in presheaf comonads

In this section, we give an explicit calculation of the values of the cosemantics functor from monads to comonads. As a first step towards this, we observe that:

Proposition 5.1 *The cosemantics functor* $\mathrm{Cosem}\colon \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}} \to \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et})$ *sends every monad to a presheaf comonad; whence it admits a factorisation to within isomorphism through* $\mathsf{Q}_{(-)}\colon \mathcal{C}\mathrm{of} \to \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et})$.

Proof. Note that the second clause follows from the first and Proposition 3.7. To prove the first, let $\mathsf{T} \in \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})$. To show that $\mathrm{Cosem}(\mathsf{T})$ is a presheaf comonad, it suffices by Proposition 3.2 to prove that its underlying endofunctor preserves connected limits. Since this endofunctor is engendered by ${}^{\mathsf{T}}U\colon {}^{\mathsf{T}}\mathcal{S}\mathrm{et} \to \mathcal{S}\mathrm{et}$ and its (limit-preserving) right adjoint, it suffices to show that ${}^{\mathsf{T}}U$ preserves connected limits. In fact, it creates them: for indeed, since a T-comodel $\boldsymbol{S}$ in $\mathcal{S}\mathrm{et}$ involves co-operations $[\![t]\!]\colon S \to A \times S$ subject to suitable equations, and since each functor of the form $A \times (-)$ preserves connected limits, it follows easily that, for any connected diagram of T-comodels, the limit of the diagram of underlying sets bears a unique comodel structure making it the limit in the category of comodels. □

What we would like to do is to give an explicit description of the factorisation of this proposition. Thus, at the level of objects, we will describe for each accessible monad T on $\mathcal{S}\mathrm{et}$ a small category $\mathbb{B}_{\mathsf{T}}$ such that $\mathrm{Cosem}(\mathsf{T}) \cong \mathsf{Q}_{\mathbb{B}_{\mathsf{T}}}$; or equally, in light of Proposition 3.10, such that we have an isomorphism in $\mathcal{C}\mathrm{AT}/\mathcal{S}\mathrm{et}$:

$$\begin{array}{ccc} {}^{\mathsf{T}}\mathcal{S}\mathrm{et} & \xrightarrow{\ \cong\ } & \mathbb{B}_{\mathsf{T}}\text{-}\mathcal{S}\mathrm{et}\,. \\ & {}^{\mathbb{T}}U \searrow \quad \swarrow U^{\mathbb{B}_{\mathsf{T}}} & \\ & \mathcal{S}\mathrm{et} & \end{array} \tag{23}$$

We term this category $\mathbb{B}_{\mathsf{T}}$ the *behaviour category* of T.

5.2. Behaviours and the final comodel

The first step is to describe the object-set of the behaviour category $\mathbb{B}_{\mathsf{T}}$ associated with an accessible monad T. By considering (23), we see that this object-set can be found as the image under $U^{\mathbb{B}_{\mathsf{T}}}$ of the final object of $\mathbb{B}_{\mathsf{T}}$-$\mathcal{S}\mathrm{et}$: and so equally as the underlying set of the *final comodel* of T. While there are many possible constructions of the final comodel – see, for example, [32, Theorem 2.2] or [27, Lemma 4.6] – we would like to give a new one which fully exploits the fact that the structures we are working with are comodels.

As is well known, when looking at coalgebraic structures, characterising the final object is bound up with answering the question of when states have the same observable behaviour. For example, if $(g, n)\colon S \to V \times S$ and $(g', n')\colon S' \to V \times S'$ are comodels of the theory of V-valued input, we may say that states $s \in S$ and $s' \in S'$ are *behaviourally equivalent* if they yield the same stream of values:

$$(g(s), g(n(s)), g(n(n(s))), \dots) = (g'(s'), g'(n'(s')), g'(n'(n'(s'))), \dots)\,.$$

We may restate this property in more structural ways. Indeed, states $s \in S$ and $s' \in S'$ are behaviourally equivalent just when any of the following conditions holds:

- They are related by some *bisimulation*, i.e., a relation $R \subseteq S \times S'$ whose projection morphisms $S \leftarrow R \rightarrow S'$ can be lifted to a span of comodels $\boldsymbol{S} \leftarrow \boldsymbol{R} \rightarrow [\boldsymbol{S}']$.
- They become equal in some comodel $\boldsymbol{S}''$; i.e., we can find a cospan of comodel homomorphisms $q\colon \boldsymbol{S} \to \boldsymbol{S}'' \leftarrow \boldsymbol{S}'\colon q'$ such that $q(s) = q'(s')$;
- They become equal in the final comodel $\boldsymbol{V}^{\mathbb{N}}$.

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

The correspondence between these conditions holds in much greater generality; see, for example [35]. However, for the comodels of an accessible monad T, there is a further, yet more intuitive, formulation: s and s' are behaviourally equivalent if, in running any T-computation $t \in T(A)$, we obtain the same A-value by running t with $\boldsymbol{S}$ from initial state s, as by running t with $\boldsymbol{S}'$ from initial state s'. More formally, we have the following definition, which appears to be novel – though it is closely related to [26]'s notion of *comodel bisimulation*.

Definition 5.2 (Behaviour of a state). *Let* T *be an accessible monad and* $\boldsymbol{S}$ *a* T*-comodel. The* behaviour β_s *of a state* $s \in S$ *is the family of functions*

$$(\beta_s)_A \colon T(A) \to A \qquad\qquad t \mapsto \pi_1([\![t]\!]_{\boldsymbol{S}}(s)) \,.$$

Given T*-comodels* $\boldsymbol{S}$ *and* $\boldsymbol{S}'$*, we say that states* $s \in S$ *and* $s' \in S'$ *are* operationally equivalent *(written* $s \sim_o s'$*) if* $\beta_s = \beta_{s'}$.

We now show that operational equivalence has the same force as the other notions of behavioural equivalence listed above.

Proposition 5.3 *Let* T *be an accessible monad and let* $\boldsymbol{S}, \boldsymbol{S}'$ *be* T*-comodels in* $\mathcal{S}\mathrm{et}$. *For any states* $s \in S$ *and* $s' \in S'$*, the following conditions are equivalent:*

(i) s *and* s' *are operationally equivalent;*
(ii) $s \, R \, s'$ *for some bisimulation* $R \subseteq S \times S'$ *between* $\boldsymbol{S}$ *and* $\boldsymbol{S}'$*;*
(iii) $q(s) = q'(s')$ *for some cospan of homomorphisms* $q \colon \boldsymbol{S} \to \boldsymbol{S}'' \leftarrow \boldsymbol{S}' \colon q'$*;*
(iv) $f(s) = f'(s')$ *for* $f \colon \boldsymbol{S} \to \boldsymbol{B}_{\mathsf{T}} \leftarrow \boldsymbol{S}' \colon f'$ *the unique morphisms to the final comodel.*

Proof. For (i) $\Rightarrow$ (ii), we show that operational equivalence $\sim_o$ is a bisimulation between $\boldsymbol{S}$ and $\boldsymbol{S}'$; this means showing that, if $u_1 \sim_o u_2$ and $t \in T(A)$, then the co-operations $[\![t]\!]_{\boldsymbol{S}}(s_1) = (a_1, {s_1}')$ and $[\![t]\!]_{\boldsymbol{S}'}(s_2) = (a_2, {s_2}')$ satisfy $a_1 = a_2 \in A$ and ${s_1}' \sim_o {s_2}' \in S$. We have $a_1 = a_2$ since $s_1 \sim_o s_2$. To show ${s_1}' \sim_o {s_2}'$, consider any term $u \in T(B)$, and observe that by (the dual of) Lemma 2.13 we have

$$[\![t(\lambda a.\, u)]\!](s_1) = [\![u]\!]({s_1}') \;\text{ and }\; [\![t(\lambda a.\, u)]\!](s_2) = [\![u]\!]({s_2}') \,.$$

Since $s_1 \sim_o s_2$, the left-hand sides above have the same first component; whence the same is true for the right-hand sides, so that ${s_1}' \sim_o {s_2}'$ as desired.

The next two implications are standard. For (ii) $\Rightarrow$ (iii), we take $\boldsymbol{S} \to \boldsymbol{S}'' \leftarrow \boldsymbol{S}'$ to be the pushout of $\boldsymbol{S} \leftarrow \boldsymbol{R} \to \boldsymbol{S}'$; and for (iii) $\Rightarrow$ (iv), we postcompose $\boldsymbol{S} \to \boldsymbol{S}'' \leftarrow \boldsymbol{S}'$ with the unique comodel map $\boldsymbol{S}'' \to \boldsymbol{B}_{\mathsf{T}}$. Finally, for (iv) $\Rightarrow$ (i), note by the definition of comodel homomorphism that if we have $h \colon \boldsymbol{S} \to \boldsymbol{S}'$ then $\beta_s = \beta_{h(s)}$ for all $s \in S$. So if $f(s) = f'(s')$ as in (iv) then $\beta_s = \beta_{f(s)} = \beta_{f'(s')} = \beta_{s'}$ and so $s \sim_o s'$ as desired. □

From this result, we see that a final T-comodel can have at most one element of a given behaviour β. In fact, in the spirit of [21, Theorem 4], we may characterise the final comodel as having exactly one element of each behaviour β which is *admissible*, in the sense of being the behaviour of some element of some comodel. It turns out that this requirement can be captured purely algebraically.

Notation 5.4 Let T be an accessible monad. Given terms $t \in T(A)$ and $u \in T(B)$, we write $t \gg u$ for the term $t(\lambda a.\, u) \in T(B)$. Noting that $\gg$ is an associative operation, we may write $t \gg u \gg v$ for $(t \gg u) \gg v = t \gg (u \gg v)$, and so on. Given $t \in T(A)$, we also write $\tilde{t} \in T1$ for the unary term $t(\lambda a.\, *)$ (where $*$ is the unique element of 1). Note that $\tilde{t}$ and $t \gg (-)$ are interdefinable via

$$\tilde{t} = t \gg \mathrm{id} \qquad \text{and} \qquad t \gg u = \tilde{t}(u) \,.$$

The intuition is that, if t and u are programs returning values in A and B, then $t \gg u$ is the program which first performs t, then discards the return value and continues as u. The notation we use is borrowed from Haskell, where `t >> u` is used with exactly this sense.

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

Definition 5.5 (Admissible behaviour). *By an* admissible behaviour *for* T*, we mean a natural transformation* $\beta : T \Rightarrow \mathrm{id}_{\mathrm{Set}}$ *whose components satisfy the equation*

$$\beta_A(t(u)) = \beta_A(t \gg u_{\beta_B(t)}) \qquad \text{for all } t \in TB \text{ and } u \in (TA)^B. \tag{24}$$

We may drop the subscript in "β_A*" where this does not lead to ambiguity.*

The condition in (24) is intuitively reasonable: it says that, if the result of running $t \in T(B)$ is $b \in B$, then the result of running $t(u) \in T(A)$ coincides with that of running t, discarding the return value, and then running u_b.

As for naturality of β, this says in more elementary terms that, for any function $f : B \to A$ and any $t \in T(B)$, we have $\beta_A(t(\lambda b.\, f(b))) = f(\beta_B(t))$. In particular, when f is the function constant at $a \in A$, this reduces to the requirement that

$$\beta_A(t \gg a) = a \qquad \text{for all } t \in T(B) \text{ and } a \in A\,; \tag{25}$$

and in fact, this suffices to recover the full naturality, since by applying (24) and (25) in succession we obtain $\beta_A(t(\lambda b.\, f(b))) = \beta_A(t \gg f(\beta_B(t))) = f(\beta_B(t))$.

Remark 5.6 *If we have a presentation of the accessible monad* T *by an algebraic theory* $\mathbb{T}$*, then we can use (24) and induction on the structure of* $\mathbb{T}$*-terms to show that an admissible behaviour* β *is determined by the values* $\beta(\sigma_1 \gg \cdots \gg \sigma_n) \in |\sigma_n|$ *for each non-empty list* $\sigma_1, \ldots, \sigma_n$ *of generating operations in* Σ *(where each operation* σ_i *is identified with the corresponding term* $\sigma_i(\lambda a.\, a) \in T(|\sigma_i|)$*). This is practically useful in computing the admissible behaviours of a theory.*

We now describe the final comodel of T. The key to doing will be the fact, as expressed by the following lemma, that a general comodel is completely determined by the behaviours associated with each state together with the action on states induced by each unary operation of T.

Notation 5.7 *Given a* T*-comodel* $\boldsymbol{S}$*, a state* $s \in S$ *and a unary operation* $m \in T(1)$*, the cointerpretation* $[\![m]\!]_{\boldsymbol{S}}(s)$ *is an element of* $1 \times S$*; by abuse of notation, we identify this element with its second projection in* S*.*

Lemma 5.8 *Let* $\boldsymbol{S}$ *be a* T*-comodel. For any* $t \in T(A)$ *and* $s \in S$ *we have*

$$[\![t]\!]_{\boldsymbol{S}}(s) = (\beta_s(t), [\![\tilde{t}]\!]_{\boldsymbol{S}}(s))$$

where β_s *is the behaviour of the state* s*.*

Proof. By definition of β_s, the first component of $[\![t]\!]_{\boldsymbol{S}}(s)$ is $\beta_s(t)$. As for the second component $s' \in S$, if we write $*$ for the unique element of 1, then it follows from Lemma 2.13(i) that $[\![\tilde{t}\,]\!]_{\boldsymbol{S}}(s) = [\![t(\lambda a.\ *)]\!]_{\boldsymbol{S}}(s) = [\![*]\!]_{\boldsymbol{S}}(s') = (*, s')$, so that under the convention of Notation 5.7 we have $s' = [\![\tilde{t}\,]\!]_{\boldsymbol{S}}(s)$. □

Proposition 5.9 *The final comodel* $\boldsymbol{B}_{\mathsf{T}}$ *of an accessible monad* T *is the set of admissible behaviours with* $[\![t]\!]_{\boldsymbol{B}_{\mathsf{T}}}(\beta) = (\beta(t), \partial_t\beta)$*, where* $\partial_t\beta(u) = \beta(t \gg u)$*.*

Proof. Since every accessible monad can be presented by some algebraic theory, it follows from Remark 5.6 that the admissible behaviours of T do indeed form a set. We must also show $[\![t]\!]$ is well-defined, i.e., that if β is admissible, then so is $\partial_t\beta$. First, we verify naturality of $\partial_t\beta$, which by the above discussion will follow from (25) for $\partial_t\beta$, which holds since:

$$(\partial_t\beta)(u \gg a) = \beta(t \gg (u \gg a)) = \beta((t \gg u) \gg a) = a\,.$$

We now verify (24), which holds by the calculation that:

$$(\partial_t\beta)(u(v)) = \beta(t \gg u(v)) = \beta((t \gg u)(v)) = \beta(t \gg u \gg v_{\beta(t \gg u)}) = (\partial_t\beta)(u \gg v_{\partial_t\beta(u)})\,.$$

So ∂_t is a well-defined operation on admissible behaviours. We now show $\boldsymbol{B}_{\mathsf{T}}$ is a comodel, i.e., that the conditions of (10) hold. For the first condition $[\![a]\!]_{\boldsymbol{B}_{\mathsf{T}}} = \nu_a$, we must show that

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

$(\beta(a), \partial_a\beta) = (a, \beta)$ for all $\beta \in B_\mathsf{T}$: but $\beta(a) = a$ by (24), while that $\partial_a\beta = \beta$ is clear since $a \gg (-)$ is the identity operator. For the second condition in (10), we must show for any $\beta \in B_\mathsf{T}$ that $[\![t(u)]\!](\beta) = (\beta(t(u)), \partial_{t(u)}\beta)$ is equal to

$$\langle [\![u_a]\!] \rangle_{a\in A}([\![t]\!](\beta)) = \langle [\![u_a]\!] \rangle_{a\in A}(\beta(t), \partial_t\beta) = [\![u_{\beta(t)}]\!](\partial_t\beta) = (\partial_t\beta(u_{\beta(t)}), \partial_{u_{\beta(t)}}(\partial_t\beta)) \,.$$

But in the first component $\beta(t(u)) = \beta(t \gg u_{\beta(t)}) = \partial_t\beta(u_{\beta(t)})$; while in the second,

$$\begin{aligned}\partial_{t(u)}\beta(v) &= \beta(t(u) \gg v) = \beta(t(\lambda a.\, u_a \gg v)) \\ &= \beta(t \gg u_{\beta(t)} \gg v) = \partial_t\beta(u_{\beta(t)} \gg v) = (\partial_{u_{\beta(t)}}(\partial_t\beta))(v)\end{aligned}$$

as desired. So $\boldsymbol{B}_\mathsf{T}$ is a comodel.

We now show that, for any comodel $\boldsymbol{S}$, there is a homomorphism $\beta_{(-)} \colon \boldsymbol{S} \to \boldsymbol{B}_\mathsf{T}$ given by $s \mapsto \beta_s$. For this to be well-defined, each β_s must be an admissible behaviour. By Lemmas 5.8 and 2.13(i), we have for any $t \in T(B)$ and $u \in T(A)^B$ that

$$[\![t(u)]\!]_\boldsymbol{S}(s) = [\![u_{\beta_s(t)}]\!]_\boldsymbol{S}([\![\tilde{t}\,]\!]_\boldsymbol{S}(s)) = [\![\tilde{t}.u_{\beta_s(t)}]\!]_\boldsymbol{S}(s) = [\![t \gg u_{\beta_s(t)}]\!]_\boldsymbol{S}(s)$$

which on taking first components gives (24). Moreover, when u above is constant at a bare variable $a \in A$, we have by Lemma 2.13(i) that

$$[\![t \gg a]\!]_\boldsymbol{S}(s) = [\![a]\!]_\boldsymbol{S}([\![\tilde{t}\,]\!]_\boldsymbol{S}(s)) = (a, [\![\tilde{t}\,]\!]_\boldsymbol{S}(s))$$

so that on first components we have $\beta_s(t \gg a) = a$ as required for (25).

We now show that $\beta_{(-)}$ is a homomorphism $\boldsymbol{S} \to \boldsymbol{B}_\mathsf{T}$; for which we calculate

$$\begin{aligned}(1 \times \beta_{(-)})[\![t]\!]_\boldsymbol{S}(s) &= (1 \times \beta_{(-)})(\beta_s(t), [\![\tilde{t}\,]\!]_\boldsymbol{S}(s)) = (\beta_s(t), \beta_{[\![\tilde{t}\,]\!]_\boldsymbol{S}(s)}) \\ &= (\beta_s(t), \partial_t(\beta_s)) = [\![t]\!]_{\boldsymbol{B}_\mathsf{T}}(\beta_s) \,.\end{aligned}$$

It remains to show that $\beta_{(-)}$ is the *unique* homomorphism $\boldsymbol{S} \to \boldsymbol{B}_\mathsf{T}$. But since $[\![t]\!]_{\boldsymbol{B}_\mathsf{T}}(\beta) = (\beta(t), \partial_t\beta)$, the behaviour of any $\beta \in \boldsymbol{B}_\mathsf{T}$ is β itself; and since as in Proposition 5.3, homomorphisms preserve behaviour, any homomorphism $\boldsymbol{S} \to \boldsymbol{B}_\mathsf{T}$ must necessarily send s to β_s. □

Example 5.10 *While the final comodels of the algebraic theories considered so far are well known, we illustrate the construction via admissible behaviours, exploiting Remark 5.6 to compute them in each case.*

- *For V-valued input, an admissible behaviour β is determined by the values $W_n := \beta((\mathsf{read} \gg)^n\mathsf{read}) \in V$ for each $n \in \mathbb{N}$. But since the theory has no equations, any such choice of values $W \in V^\mathbb{N}$ yields an admissible behaviour. Thus, the final comodel is $V^\mathbb{N}$, and we can read off from Proposition 5.9 that $[\![\mathsf{read}]\!] \colon V^\mathbb{N} \to V \times V^\mathbb{N}$ is given by $W \mapsto (W_0, \partial W)$, where $(\partial W)_i = W_{i+1}$.*
- *For V-valued output, an admissible behaviour β is determined by the trivial choices $\beta(\mathsf{put}_{v_1} \gg \cdots \gg \mathsf{put}_{v_n}) \in 1$; whence there is a unique admissible behaviour, and the final comodel is the one-element set with the trivial co-operations.*
- *For V-valued read-only state, since $\mathsf{get} \gg (-)$ is the identity operator, an admissible behaviour is uniquely determined by the value $\beta(\mathsf{get}) \in V$. Any such choice yields an admissible behaviour, and so the final comodel is V with the co-operation $[\![\mathsf{get}]\!] = \Delta \colon V \to V \times V$.*
- *For V-valued state, $\mathsf{get} \gg (-)$ is again the identity operator, and so an admissible behaviour β is determined by the values $\beta(\mathsf{put}_{v_1} \gg \cdots \gg \mathsf{put}_{v_n} \gg \mathsf{get}) \in V$ for $v_1, \ldots, v_n \in V$. When $n > 0$, the put axioms force $\beta(\mathsf{put}_{v_1} \gg \cdots \gg \mathsf{put}_{v_n} \gg \mathsf{get}) = v_n$ and so β is uniquely determined by $\beta(\mathsf{get}) \in V$. Thus, again the final comodel is V, with the same $[\![\mathsf{get}]\!]$ as before, and with $[\![\mathsf{put}]\!] = \pi_1 \colon V \times V \to V$.*

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

5.3. The behaviour category of an accessible monad

We saw in Lemma 2.13 above that if T is an accessible monad, then any T-comodel $\boldsymbol{S}$ in $\mathcal{S}\mathrm{et}$ is completely determined by the two functions

$$\begin{array}{ll} S \to B_{\mathsf{T}} & S \times T(1) \to S \\ s \mapsto \beta_s & (s, m) \mapsto [\![m]\!]_{\boldsymbol{S}}(s)\,. \end{array} \tag{26}$$

giving the behaviour of each state, together with what we might call the *dynamics* of the comodel: the right action of unary operations on states. However, these two structures are not independent. One obvious restriction is that the right action by $m \in T(1)$ must send elements of behaviour β to elements of behaviour $\partial_m\beta$. However, due to (24) there is a further constraint; the following definition is intended to capture this.

Definition 5.11 (β-equivalence). *Let T be an accessible monad, and let β be an admissible T-behaviour. We say that unary operations $m, n \in T(1)$ are* atomically β-equivalent *if there exists $v \in T(A)$ and $m', n' \in T(1)^A$ such that*

$$m = v(m') \quad \text{and} \quad n = v(n') \quad \text{and} \quad {m'}_{\beta(v)} = {n'}_{\beta(v)}\,. \tag{27}$$

We write $\sim_\beta$ for the smallest equivalence relation on $T(1)$ which identifies atomically β-equivalent terms. Alternatively, $\sim_\beta$ is the smallest equivalence relation such that $t(m) \sim_\beta (t \gg m_{\beta(t)})$ for all $t \in T(A)$ and $m \in T(1)^A$.

Remark 5.12 *If we have a presentation of the accessible monad T by an algebraic theory $\mathbb{T} = (\Sigma, \mathcal{E})$, then we may simplify the task of computing the equivalence relation $\sim_\beta$ by observing that, by induction on the structure of $\mathbb{T}$-terms, each $m \in T(1)$ is $\sim_\beta$-equivalent to $\sigma_1 \gg \cdots \gg \sigma_n \gg \mathrm{id}$ for some $\sigma_1, \ldots, \sigma_n \in \Sigma$.*

The motivation for this definition is that $\sim_\beta$ will identify two unary operations if and only if they act in the same way on any state of behaviour β. The "only if" direction is part (iii) of the following lemma; the "if" will be proved in Corollary 5.18.

Notation 5.13 If $m \in T(1)$ and $t \in T(A)$, we write $m.t$ for the substitution $m(t)$.

Lemma 5.14 *Let β be an admissible behaviour of the accessible monad T, and let $m, n, p \in T(1)$.*

(i) If $m \sim_\beta n$ then $m.p \sim_\beta n.p$;

(ii) If $m \sim_{\partial_p\beta} n$ then $p.m \sim_\beta p.n$;

(iii) If $s \in \boldsymbol{S}$ is a state of behaviour β and $m \sim_\beta n$, then $[\![m]\!]_{\boldsymbol{S}}(s) = [\![n]\!]_{\boldsymbol{S}}(s)$.

Proof. For (i), if m and n are atomically β-equivalent via v, m', n', then $m.p$ and $n.p$ are so via v, $({m'}_a.p : a \in A)$ and $({n'}_a.p : a \in A)$; whence $m \sim_\beta n$ implies $m(p) \sim_\beta n(p)$. (ii) is similar, observing that if m,n are atomically $\partial_p\beta$-equivalent via v, m', n', then $p.m$ and $p.n$ are atomically β-equivalent via $p.v, m', n'$. Finally, for (iii), if m, n are atomically β-equivalent, then writing $s' = [\![v \gg \mathrm{id}]\!](s)$ we have

$$[\![m]\!](s) = [\![v(m')]\!](s) = [\![{m'}_{\beta(v)}]\!](s') = [\![{n'}_{\beta(v)}]\!](s') = [\![v(n')]\!](s) = [\![n]\!](s)\,. \qquad \square$$

With this in place, we can now give the main definition of this section:

Definition 5.15 (Behaviour category of a monad). *Let T be an accessible monad. The* behaviour category $\mathbb{B}_{\mathsf{T}}$ *of T has admissible behaviours as objects, and hom-sets*

$$\mathbb{B}_{\mathsf{T}}(\beta, \beta') = \{m \in T(1) \mid \beta' = \partial_m\beta\}/{\sim_\beta}\,.$$

Identities are given by the neutral element of $T(1)$, and the composite of $m\colon \beta \to \beta'$ and $n\colon \beta' \to \beta''$ is $m.n\colon \beta \to \beta''$; note this is well-defined by Lemma 5.14(i–ii).

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

Theorem 5.16 *Given an accessible monad* T *with behaviour category* $\mathbb{B}_\mathsf{T}$*, the category of* T*-comodels is isomorphic to the category of left* $\mathbb{B}_\mathsf{T}$*-sets via an isomorphism commuting with the forgetful functors to* $\mathcal{S}\mathrm{et}$*:*

$$\begin{array}{ccc} {}^{\mathsf{T}}\mathcal{S}\mathrm{et} & \xrightarrow{\cong} & \mathbb{B}_\mathsf{T}\text{-}\mathcal{S}\mathrm{et}\,. \\ & {}^{\mathsf{T}}U \searrow \quad \swarrow U^{\mathbb{B}_\mathsf{T}} & \\ & \mathcal{S}\mathrm{et} & \end{array} \tag{28}$$

For a comodel $\boldsymbol{S}$*, the corresponding left* $\mathbb{B}_\mathsf{T}$*-set* $X_{\boldsymbol{S}}$ *has underlying set* S*, projection to* B_T *given by the behaviour map* $\beta_{(-)}\colon S \to B_\mathsf{T}$*, and action given by*

$$(s, \beta_s \xrightarrow{m} \partial_m \beta_s) \mapsto [\![m]\!]_{\boldsymbol{S}}(s)\,; \tag{29}$$

for a left $\mathbb{B}_\mathsf{T}$*-set* $(X, p, *)$*, the corresponding comodel* $\boldsymbol{S}_X$ *has underlying set* X *and*

$$[\![t]\!]_{\boldsymbol{S}_X}\colon x \mapsto (\beta(t), \tilde{t} * x) \qquad \text{for all } x \in p^{-1}\beta\,. \tag{30}$$

Proof. We concentrate on giving the isomorphism of categories at the level of objects; indeed, since it is to commute with the faithful functors to $\mathcal{S}\mathrm{et}$, to obtain the isomorphism on arrows we need only check that a function lifts along ${}^{\mathsf{T}}U$ just when it lifts along $U^{\mathbb{B}_\mathsf{T}}$, and we leave this to the reader.

Now, on objects, one direction is easy: the action map (29) of the presheaf associated with a comodel is well defined by Lemma 5.14(iii) and is clearly functorial. In the converse direction, we must prove that the $\boldsymbol{S}_X$ associated with a left $\mathbb{B}_\mathsf{T}$-set X satisfies the comodel axioms in (10). For the first axiom, we have that $[\![a]\!](s) = (\beta(a), \tilde{a} * s) = (a, s) = \nu_a(s)$ for all $s \in p^{-1}\beta$, since $\beta(a) = a$ and $\tilde{a} = \mathrm{id} \in T(1)$. For the second, given $s \in p^{-1}\beta$ we must show $[\![t(u)]\!](s) = (\beta(t(u)), \widetilde{t(u)} * s)$ is equal to

$$\langle [\![u_a]\!] \rangle_{a\in A}([\![t]\!](s)) = \langle [\![u_a]\!] \rangle(\beta(t), \tilde{t} * s) = [\![u_{\beta(t)}]\!](\tilde{t} * s) = (\partial_t\beta(u_{\beta(t)}), \tilde{u}_{\beta(t)} * (\tilde{t} * s))\,.$$

But in the first component $\beta(t(u)) = \beta(t \gg u_{\beta(t)}) = \partial_t\beta(u_{\beta(t)})$; while in the second, since we have $\tilde{u}_{\beta(t)} * (\tilde{t} * s) = (\tilde{u}_{\beta(t)} \circ \tilde{t}) * s = \tilde{t}(\tilde{u}_{\beta(t)}) * s$, it suffices to show that we have $\tilde{t}(\tilde{u}_{\beta(t)}) = \widetilde{t(u)}\colon \beta \to \partial_t\beta$ in $\mathbb{B}_\mathsf{T}$; but

$$\tilde{t}(\tilde{u}_{\beta(t)}) = t(\lambda a.\, \tilde{u}_{\beta(t)}) \quad \text{and} \quad \widetilde{t(u)} = t(\lambda a.\, \tilde{u}_a)$$

and these two terms are clearly atomically β-equivalent, and so equal as morphisms in $\mathbb{B}_\mathsf{T}$. This shows that $\boldsymbol{S}_X$ is a T-comodel.

It remains to show that these two assignments are mutually inverse. For any comodel $\boldsymbol{S}$, the comodel $\boldsymbol{S}_{X_{\boldsymbol{S}}}$ clearly has the same underlying set, but also the same comodel structure, since

$$[\![t]\!]_{\boldsymbol{S}_{X_{\boldsymbol{S}}}}(s) = (\beta_s(t), \tilde{t} * s) = (\beta_s(t), [\![t \gg \mathrm{id}]\!]_{\boldsymbol{S}}(s)) = [\![t]\!]_{\boldsymbol{S}}(s)\,.$$

On the other hand, for any $\mathbb{B}_\mathsf{T}$-set X, the $\mathbb{B}_\mathsf{T}$-set $X_{\boldsymbol{S}_X}$ has the same underlying set, but also the same projection to B_T, since (30) exhibits each $x \in \boldsymbol{S}_X$ as having behaviour $p(x)$; and the same action map, since $m *_{X_{\boldsymbol{S}_X}} s = [\![m]\!]_{\boldsymbol{S}_X}(s) = m *_X s$. □

Corollary 5.17 *Let* T *be an accessible monad. For each* T*-admissible behaviour* β*, there exists a comodel* $\boldsymbol{\beta}$ *freely generated by a state of behaviour* β*; it has underlying set* $T(1)/{\sim_\beta}$*, co-operations* $[\![t]\!]_{\boldsymbol{\beta}}(m) = (\beta(m.t), m.\tilde{t})$*, and generating state* $\mathrm{id} \in \boldsymbol{\beta}$*. Morphisms* $\beta \to \beta'$ *in the behaviour category of* T *are in functorial bijection with comodel homomorphisms* $\boldsymbol{\beta}' \to \boldsymbol{\beta}$*.*

Proof. We obtain $\boldsymbol{\beta}$ as the image of the representable functor $\mathbb{B}(\beta, -)$ under the equivalences of (19) and (28). The final clause follows from the Yoneda lemma. □

We now tie up a loose end by proving the converse to Lemma 5.14(iii).

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

Corollary 5.18 *Let* T *be an algebraic theory,* β *an admissible* T*-behaviour, and* $m, n \in T(1)$*. We have* $m \sim_\beta n$ *if, and only if, for every* T*-comodel* $\boldsymbol{S}$ *and state* $s \in \boldsymbol{S}$ *of behaviour* β*, we have* $[\![m]\!]_{\boldsymbol{S}}(s) = [\![n]\!]_{\boldsymbol{S}}(s)$*.*

Proof. The "only if" direction is Lemma 5.14(iii). For the "if" direction, consider the T-comodel β classifying states of behaviour β and the state of β given by $\mathrm{id} \in T(1)/{\sim_\beta}$. By our assumption (and Notation 5.7), we have that $m = [\![m]\!]_\beta(\mathrm{id}) = [\![n]\!]_\beta(\mathrm{id}) = n$ in the underlying set $T(1)/{\sim_\beta}$ of β, i.e., $m \sim_\beta n$ as desired. □

Remark 5.19 *In computing the comodel* β *classifying an admissible behaviour* β*, the main problem is to determine suitable equivalence-class representatives for elements of the underlying set* $T(1)/{\sim_\beta}$*. Suppose we have a subset* $\{\mathrm{id}\} \subseteq S \subseteq T(1)$ *which we believe constitutes a set of such representatives. By Corollary 5.17, a necessary condition for this belief to be correct is that* S *underlie a comodel* $\boldsymbol{S}$ *with*

$$[\![t]\!]_{\boldsymbol{S}}(s) = (\beta(s.t), s') \qquad \text{for some } s' \in S \text{ with } s' \sim_\beta s.\tilde{t}\,; \tag{31}$$

of course, this $\boldsymbol{S}$ *will then be the desired classifier for states of behaviour* β*.*

In fact, the preceding result tells us that this necessary condition is also *sufficient*. Indeed, if $S \subseteq T(1)$ bears a comodel structure satisfying (31), then for each $m \in T(1)$ we have $[\![m]\!]_{\boldsymbol{S}}(\mathrm{id}) \in S$ in the $\sim_\beta$-equivalence class of m. So S contains at least one element from each $\sim_\beta$-classes of $T(1)$. In fact, it contains *precisely* one such element. For indeed, if $s \neq s' \in S$, then $[\![s]\!]_{\boldsymbol{S}}(\mathrm{id}) = s \neq s' = [\![s']\!]_{\boldsymbol{S}}(\mathrm{id})$, whence $s \nsim_\beta s'$ by Corollary 5.18.

Example 5.20 *For each of our running examples of algebraic theories, we compute the comodels classifying each admissible behaviour, and so the behaviour category. For these examples, it is simple enough to find the classifying comodels directly without exploiting Remark 5.19.*

- *For* V*-valued input, the object-set of the behaviour category is* $V^{\mathbb{N}}$*, and for each behaviour* $W \in V^{\mathbb{N}}$*, the comodel* $\boldsymbol{W}$ *classifying this behaviour may be taken to have underlying set* $\mathbb{N}$ *with co-operation* $[\![\mathsf{read}]\!](n) = (W_n, n+1)$*; the universal state of behaviour* W *is then* $0 \in \boldsymbol{W}$*. Morphisms* $W \to W'$ *in the behaviour category correspond to states of* $\boldsymbol{W}$ *of behaviour* W'*, and these can be identified with natural numbers* i *such that* ${W'}_n = W_{n+i}$ *for all* $k \in \mathbb{N}$*.*
- *For* V*-valued output, the behaviour category has a single object* $*$*, and the comodel* $\boldsymbol{V}^*$ *classifying this unique behaviour has as underlying set the free monoid* V^**, with co-operations* $[\![\mathsf{write}_v]\!](W) = Wv$*; the universal state is the empty word* $\varepsilon \in \boldsymbol{V}^*$*. Endomorphisms of* $*$ *in the behaviour category correspond to states of* $\boldsymbol{V}^*$*, so that the behaviour category is precisely the the one-object category corresponding to the monoid* V^**.*
- *For* V*-valued read-only state, the behaviour category has object-set* V*, and the comodel classifying* $v \in V$ *is the one-element comodel* $\boldsymbol{v}$ *with* $[\![\mathsf{get}]\!](*) = (v, *)$*. Clearly, there are no non-identity homomorphisms between such comodels, so that the behaviour category is the discrete category on the set* V*.*
- *For* V*-valued state, the behaviour category again has object-set* V*, while the comodel* $\boldsymbol{v}$ *classifying any* $v \in V$ *is the final comodel* $\boldsymbol{V}$*, with universal state* v*. Morphisms* $v \to v'$ *in the behaviour category thus correspond to comodel homomorphisms* $\boldsymbol{V} \to \boldsymbol{V}$*, and since* $\boldsymbol{V}$ *is final, the only such is the identity. Thus, the behaviour category is the codiscrete category* ∇V *on* V*, with a unique arrow between every two objects.*

5.4. Functoriality

By Theorem 5.16, the assignment $\mathsf{T} \mapsto \mathbb{B}_{\mathsf{T}}$ is the action on objects of the desired factorisation of the cosemantics functor $\mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}} \to \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et})$ through $\mathsf{Q}_{(-)} \colon \mathcal{C}\mathrm{of} \to \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et})$. We conclude this section by describing the corresponding action on morphisms.

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

We start with a preliminary lemma; in its statement, recall from Example 2.27 how a monad morphism $f\colon \mathsf{T} \to \mathsf{R}$ induces a functor on comodels $f^*\colon {}^{\mathsf{R}}\mathcal{S}\mathrm{et} \to {}^{\mathsf{T}}\mathcal{S}\mathrm{et}$.

Lemma 5.21 *Let $f\colon \mathsf{T} \to \mathsf{R}$ in $\mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})$, and let $\boldsymbol{S}$ be a R-comodel. If the state $s \in \boldsymbol{S}$ has R-behaviour β, then the state $s \in f^*\boldsymbol{S}$ has T-behaviour $f^*\beta$ where $(f^*\beta)(t) = \beta(f(t))$.*

Proof. For any $t \in T(A)$, we have $\pi_1([\![t]\!]_{f^*\boldsymbol{S}}(s)) = \pi_1([\![f(t)]\!]_{\boldsymbol{S}}(s)) = \beta(f(t))$. □

Proposition 5.22 *Let T and R be accessible monads on $\mathcal{S}\mathrm{et}$. For each monad morphism $f\colon \mathsf{T} \to \mathsf{R}$, there is a cofunctor $\mathbb{B}_f\colon \mathbb{B}_{\mathsf{R}} \rightsquigarrow \mathbb{B}_{\mathsf{T}}$ which acts on objects by $\beta \mapsto f^*\beta$; and which, on morphisms, given $\beta \in \mathbb{B}_{\mathsf{R}}$, acts by sending $m\colon f^*\beta \to \partial_m(f^*\beta)$ in $\mathbb{B}_{\mathsf{T}}$ to $(\mathbb{B}_f)_\beta(m) := f(m)\colon \beta \to \partial_{f(m)}\beta$ in $\mathbb{B}_{\mathsf{R}}$.*

Proof. $f^*\colon \mathbb{B}_{\mathsf{R}} \rightsquigarrow \mathbb{B}_{\mathsf{T}}$ is well-defined on objects by the preceding lemma. For well-definedness on morphisms, we must show that $m \sim_{f^*\beta} n$ in $T(1)$ implies $f(m) \sim_\beta f(n)$ in $R(1)$. Clearly it suffices to do so when m and n are atomically $f^*\beta$-equivalent via terms $v \in T(A)$ and $m', n' \in T(1)^A$ satisfying

$$m = v(m') \qquad \text{and} \qquad n = v(n') \qquad \text{and} \qquad m'_{f^*\beta(v)} = n'_{f^*\beta(v)}\,.$$

But since $f^*\beta(v) = \beta(f(v))$, it follows that $f(v) \in R(A)$ and $f(m'_{(-)}), f(n'_{(-)}) \in R(1)^A$ witness $f(m)$ and $f(n)$ as atomically β-equivalent, as required. We must also check the three cofunctor axioms. The first holds since $f^*(\partial_{f(m)}\beta)(t) = (\partial_{f(m)}\beta)(f(t)) = \beta(f(m).f(t)) = \beta(f(m.t)) = f^*\beta(m.t) = \partial_m(f^*\beta)(t)$. The other two are immediate since f preserves substitution. □

We now prove that the cofunctor $\mathbb{B}_f$ of Proposition 5.22 does indeed describe the action on morphisms of the cosemantics functor.

Theorem 5.23 *The functor $\mathbb{B}_{(-)}\colon \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}} \to \mathcal{C}\mathrm{of}$ taking each accessible monad T to its behaviour category $\mathbb{B}_{\mathsf{T}}$, and each map of accessible monads $f\colon \mathsf{T} \to \mathsf{R}$ to the cofunctor $\mathbb{B}_f\colon \mathbb{B}_{\mathsf{R}} \rightsquigarrow \mathbb{B}_{\mathsf{T}}$ of Proposition 5.22, yields a within-isomorphism factorisation*

$$\begin{array}{ccc} & & \mathcal{C}\mathrm{of} \\ & \overset{\mathbb{B}_{(-)}}{\nearrow} \cong & \big\downarrow {\scriptstyle Q_{(-)}} \\ \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}} & \xrightarrow[\mathrm{Cosem}]{} & \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et})\,. \end{array}$$

Proof. It suffices to show that, for any map of accessible monads $f\colon \mathsf{T} \to \mathsf{R}$, the associated cofunctor $\mathbb{B}_f\colon \mathbb{B}_{\mathbb{R}} \rightsquigarrow \mathbb{B}_{\mathsf{T}}$ renders commutative the square

$$\begin{array}{ccc} {}^{\mathsf{R}}\mathcal{S}\mathrm{et} & \xrightarrow{\cong} & \mathbb{B}_{\mathsf{R}}\text{-}\mathcal{S}\mathrm{et} \\ \big\downarrow {\scriptstyle f^*} & & \big\downarrow {\scriptstyle \Sigma_{\mathbb{B}_f}} \\ {}^{\mathsf{T}}\mathcal{S}\mathrm{et} & \xrightarrow{\cong} & \mathbb{B}_{\mathsf{T}}\text{-}\mathcal{S}\mathrm{et} \end{array}$$

whose horizontal edges are the isomorphisms of Theorem 5.16, whose left edge is as in Example 2.27 and whose right edge is as in Definition 3.11. But indeed, given an R-comodel $\boldsymbol{S}$, its image around the lower composite is by Lemma 5.21 the $\mathbb{B}_{\mathsf{T}}$-set with underlying set S and projection and action morphisms

$$\begin{array}{ll} S \to B_{\mathsf{T}} & \textstyle\sum_{s \in S} T(1)/{\sim_{f^*(\beta_s)}} \to S \\ s \mapsto f^*(\beta_s) & (s, m) \mapsto [\![f(m)]\!]_{\boldsymbol{S}}(s)\,. \end{array}$$

On the other hand, the upper composite first sends $\boldsymbol{S}$ to the $\mathbb{B}_{\mathsf{R}}$-set with underlying set S and projection and action maps

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

$$S \to B_{\mathsf{R}} \qquad \sum_{s \in S} R(1)/{\sim_{\beta_s}} \to S$$
$$s \mapsto \beta_s \qquad (s, n) \mapsto [\![n]\!]_{\mathsf{S}}(s)\,;$$

and then applies $\Sigma_{\mathbb{B}_f}$, which, by the definition of $\Sigma_{(-)}$ and $\mathbb{B}_f$, yields the same $\mathbb{B}_{\mathsf{T}}$-set as above. □

Example 5.24 *Let $h\colon V \to W$ be a function, and let $f\colon \mathbb{T}_1 \to \mathbb{T}_2$ be the associated interpretation of V-valued output into W-valued state of Example 2.8. The induced cofunctor $f^*\colon \mathbb{B}_{\mathbb{T}_2} \to \mathbb{B}_{\mathbb{T}_1}$ has as domain the codiscrete category on W, and as codomain, the monoid V^* seen as a one-object category. On objects, f^* acts in the unique possible way; while on morphisms, given $w \in \mathbb{B}_{\mathbb{T}_2}$ and a map $f^*(w) \to *$ in $\mathbb{B}_{\mathbb{T}_1}$ – corresponding to an element $v_1 \ldots v_n \in V^*$ – we have $f^*_w(v_1 \ldots v_n)$ in $\mathbb{B}_{\mathbb{T}_2}$ given by the unique map $w \to v_n$.*

Example 5.25 *Let $h\colon W \to V$ be a function between sets, and let $f\colon \mathbb{T}_1 \to \mathbb{T}_2$ be the associated interpretation of V-valued read-only state into W-valued state of Example 2.9. The induced cofunctor $f^*\colon \mathbb{B}_{\mathbb{T}_2} \to \mathbb{B}_{\mathbb{T}_1}$ has as domain the codiscrete category on W, and as codomain the discrete category on V. On objects, f^* acts by $w \mapsto h(v)$, while on morphisms it acts in the unique possible way.*

6. Calculating the Costructure Functor

6.1. The behaviour category of an accessible comonad

In this section, we give an explicit calculation of the costructure functor from comonads to monads. Much as for the cosemantics functor, we will see that cosemantics takes its values in presheaf monads, and will explicitly associate to each accessible comonad Q a small category $\mathbb{B}_{\mathsf{Q}}$, the *behaviour category*, such that $\mathrm{Costr}(\mathsf{Q})$ is the presheaf monad of $\mathbb{B}_{\mathsf{Q}}$. We begin with some preliminary observations.

Notation 6.1 Let Q be an accessible comonad on $\mathcal{S}\mathrm{et}$. For each $x \in Q1$, we write $\iota_x\colon Q_x \to Q$ for the inclusion of the subfunctor with

$$Q_x(A) = \{a \in QA : (Q!)(a) = x \text{ in } Q1\}\,.$$

We also write $\varepsilon_x\colon Q_x \to 1$ and $\delta_x\colon Q_x \to Q_xQ$ for the natural transformations such that $\varepsilon_x = \varepsilon \circ \iota_x$ and $\iota_xQ \circ \delta_x = \delta \circ \iota_x$; to see that $\delta \circ \iota_x$ does indeed factor through ι_xQ, we note that, for any $a \in Q_xA$, the element $\delta(a) \in QQA$ satisfies $(Q!)(\delta(a)) = (Q!)(Q\varepsilon(\delta(a))) = (Q!)(a) = x$ so that $\delta(a) \in Q_xQA$ as desired.

Lemma 6.2 *Let Q be an accessible comonad on $\mathcal{S}\mathrm{et}$.*

(i) The inclusions $\iota_x\colon Q_x \to Q$ exhibit Q as the coproduct $\sum_{x \in Q1} Q_x$;

(ii) Any natural transformation $f\colon Q_x \to \sum_i F_i$ into a coproduct factors through exactly one coproduct injection $\nu_i\colon F_i \to \sum_i F_i$.

Proof. (i) holds as each QA is the coproduct of the Q_xA's, and coproducts in $[\mathcal{S}\mathrm{et}, \mathcal{S}\mathrm{et}]$ are componentwise. For (ii), the component $f_1\colon \{x\} \to \sum_{i \in I} F_i1$ of f clearly factors through just one ν_i; now naturality of f with respect to the unique morphisms $!\colon A \to 1$ shows that each f_A factors through just the same ν_i. □

Definition 6.3 (Behaviour category of a comonad) *Let Q be an accessible comonad on $\mathcal{S}\mathrm{et}$. The* behaviour category *of Q is the small category $\mathbb{B}_{\mathsf{Q}}$ in which:*

- *Objects are elements of $Q1$;*
- *Morphisms with domain $x \in Q1$ are natural transformations $\tau\colon Q_x \to \mathrm{id}$, and the codomain of such a τ is determined by the following factorisation, whose (unique) existence is asserted by Lemma 6.2:*

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

$$\begin{array}{ccc} Q_x & \xrightarrow{\tau^\sharp} & Q_{\mathrm{cod}(\tau)} \\ {\scriptstyle \delta_x}\downarrow & & \downarrow{\scriptstyle \iota_{\mathrm{cod}(\tau)}} \\ Q_xQ & \xrightarrow{\tau Q} & Q\,; \end{array} \tag{32}$$

- *The identity on $x \in Q1$ is $\varepsilon_x \colon Q_x \to \mathrm{id}$;*
- *Binary composition is given as follows, where $\tau^\sharp$ is the factorisation in (32):*

$$(Q_{\mathrm{cod}(\tau)} \xrightarrow{\upsilon} \mathrm{id}) \circ (Q_x \xrightarrow{\tau} \mathrm{id}) = (Q_x \xrightarrow{\tau^\sharp} Q_{\mathrm{cod}(\tau)} \xrightarrow{\upsilon} \mathrm{id})\,.$$

The axioms expressing unitality and associativity of composition follow from the familiar and easily-checked identities $\varepsilon_{\mathrm{cod}(\tau)} \circ \tau^\sharp = \tau$, $\varepsilon_x^\sharp = \mathrm{id}_{Q_x}$ and $(\upsilon \circ \tau^\sharp)^\sharp = \upsilon^\sharp \circ \tau^\sharp$.

We now show that the costructure monad associated with an accessible comonad Q is isomorphic to the presheaf monad $\mathsf{T}_{\mathbb{B}_\mathsf{Q}}$. In light of Proposition 4.13, it suffices to construct an isomorphism between $\mathsf{T}_{\mathbb{B}_\mathsf{Q}}$ and the *dual monad* Q° of Definition 4.11.

Proposition 6.4 *Let Q be an accessible comonad. For any $\tau \colon Q \to A \cdot \mathrm{id}$ and any $x \in Q1$, there is a unique $a_x \in A$ and $\tau_x \colon Q_x \to \mathrm{id}$ for which we have a factorisation*

$$\begin{array}{ccc} Q_x & \xrightarrow{\tau_x} & \mathrm{id} \\ {\scriptstyle \iota_x}\downarrow & & \downarrow{\scriptstyle \nu_{a_x}} \\ Q & \xrightarrow{\tau} & A \cdot \mathrm{id}\,. \end{array} \tag{33}$$

In this way, we obtain a monad isomorphism $\theta \colon \mathsf{Q}^\circ \cong \mathsf{T}_{\mathbb{B}_\mathsf{Q}}$ with components

$$\begin{aligned} \theta_A \colon [\mathcal{S}\mathrm{et}, \mathcal{S}\mathrm{et}](Q, A \cdot \mathrm{id}) &\to \textstyle\prod_{x \in \mathbb{B}_\mathsf{Q}} ((\mathbb{B}_\mathsf{Q})_x \times A) \\ \tau &\mapsto \lambda x.\, (\tau_x, a_x)\,. \end{aligned} \tag{34}$$

Proof. The existence and uniqueness of the factorisation (33) is a consequence of Lemma 6.2(ii); now that the induced morphisms (34) constitute a natural isomorphism follows from Lemma 6.2(i) and the Yoneda lemma. It remains to show that these morphisms are the components of a monad isomorphism $\mathsf{Q}^\circ \cong \mathsf{T}_{\mathbb{B}_\mathsf{Q}}$.

For compatibility with units, we have on the one hand that $\eta_A^{\mathsf{T}_{\mathbb{B}_\mathsf{Q}}}(a) = \lambda x.\, (\varepsilon_x, a)$. On the other hand, $\eta_A^{\mathsf{Q}^\circ}(a) = \nu_a \circ \varepsilon \colon Q \to \mathrm{id} \to A \cdot \mathrm{id}$, whose factorisation as in (33) is clearly given by $\eta_A^{\mathsf{Q}^\circ}(a) \circ \iota_x = \nu_a \circ \varepsilon_x$, so that $\theta_A(\eta_A^{\mathsf{Q}^\circ}(a)) = \lambda x.\, (\varepsilon_x, a)$ as desired.

We now show compatibility with multiplication. To this end, consider an element $\sigma \colon Q \to Q^\circ A \cdot \mathrm{id}$ of $Q^\circ Q^\circ A$. For each $x \in Q1$, we have an element $\tau_x \in Q^\circ A$ and a natural transformation $\sigma_x \colon Q_x \to \mathrm{id}$ rendering commutative the square to the left in

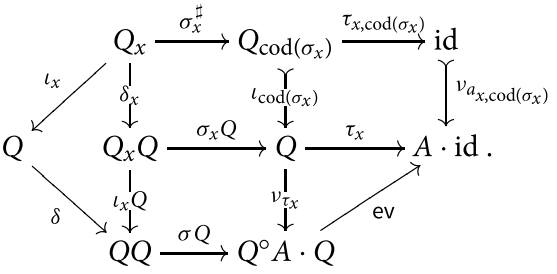
$$\begin{array}{ccc} Q_x & \xrightarrow{\sigma_x} & \mathrm{id} \\ {\scriptstyle \iota_x}\downarrow & & \downarrow{\scriptstyle \nu_{\tau_x}} \\ Q & \xrightarrow{\sigma} & Q^\circ A \cdot \mathrm{id} \end{array} \qquad \begin{array}{ccc} Q_y & \xrightarrow{\tau_{xy}} & \mathrm{id} \\ {\scriptstyle \iota}\downarrow & & \downarrow{\scriptstyle \nu_{a_{xy}}} \\ Q & \xrightarrow{\tau_x} & A \cdot \mathrm{id}\,. \end{array} \tag{35}$$

Considering now $\tau_x \in Q^\circ A$, we have for each $y \in Q1$ an element $a_{xy} \in A$ and $\tau_{xy} \colon Q_y \to \mathrm{id}$ rendering commutative the square above right. With this notation, the composite $\mu_A^{\mathsf{T}_{\mathbb{B}_\mathsf{Q}}} \circ (\theta\theta)_A$ acts on $\sigma \in Q^\circ Q^\circ A$ via

$$\sigma \xmapsto{(\theta\theta)_A} \lambda x.\, (\sigma_x, \lambda y.\, (\tau_{xy}, a_{xy})) \xmapsto{\mu_A^{\mathsf{T}_{\mathbb{B}_\mathsf{Q}}}} \lambda x.\, (\tau_{x,\mathrm{cod}(\sigma_x)} \circ \sigma_x^\sharp, a_{x,\mathrm{cod}(\sigma_x)})\,.$$

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

We must show that this is equal to the image of σ under the composite $\theta_A \circ \mu_A^{\mathsf{Q}^\circ}$. From the description of Q° in Definition 4.11, we can read off that $\mu_A^{\mathsf{Q}^\circ}(\sigma) \in Q^\circ A$ is the lower composite in the diagram



where ev is unique such that $\mathsf{ev} \circ \nu_\tau = \tau$ for all $\tau \in Q^\circ A$. To calculate the image of $\mu_A^{\mathsf{Q}^\circ}(\sigma)$ under θ_A, we observe that, in the displayed diagram, the far left region is the definition of δ_x, the two upper squares are instances of (32) and (33), the lower square is $(-)Q$ of another instance of (33), and the triangle is the definition of ev. So by unicity in (33), we have that $\theta_A(\mu_A^{\mathsf{Q}^\circ}(\sigma)) = \lambda x.\, (\tau_{x,\mathrm{cod}(\sigma_x)} \circ \sigma_x^\sharp, a_{x,\mathrm{cod}(\sigma_x)})$ as required. □

6.2. Functoriality

We now describe the manner in which the passage from an accessible comonad to its behaviour category is functorial.

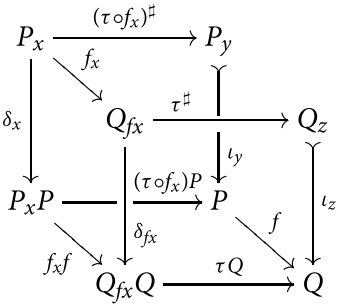
Notation 6.5 Let $f\colon \mathsf{P} \to \mathsf{Q}$ be a morphism of accessible comonads on $\mathcal{S}\mathrm{et}$ and let $x \in P1$. We write $f_x\colon P_x \to Q_{fx}$ for the unique natural transformation (whose unique existence follows from Lemma 6.2) such that $f \circ \iota_x = \iota_{fx} \circ f_x\colon P_x \to Q$.

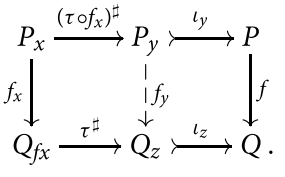
Proposition 6.6 *Each morphism $f\colon \mathsf{P} \to \mathsf{Q}$ of accessible comonads on $\mathcal{S}\mathrm{et}$ induces a cofunctor $\mathbb{B}_f\colon \mathbb{B}\mathsf{P} \rightsquigarrow \mathbb{B}\mathsf{Q}$ on behaviour categories with action on objects $f_1\colon P1 \to Q1$, and with, for each $x \in P1$, the action on homs $(\mathbb{B}\mathsf{Q})_{fx} \to (\mathbb{B}\mathsf{P})_x$ given by $\tau \mapsto \tau \circ f_x$.*

Proof. We first dispatch axiom (ii) for a cofunctor, which follows by the calculation that

$$\varepsilon^{\mathsf{Q}}_{fx} \circ f_x = \varepsilon^{\mathsf{Q}} \circ \iota_{fx} \circ f_x = \varepsilon^{\mathsf{Q}} \circ f \circ \iota_x = \varepsilon^{\mathsf{P}} \circ \iota_x = \varepsilon^{\mathsf{P}}_x\colon P_x \to \mathrm{id}.$$

We next deal with axiom (i). Let $\tau\colon Q_{fx} \to \mathrm{id}$ be an element of $(\mathbb{B}\mathsf{Q})_{fx}$ with image $\tau \circ f_x$ in $(\mathbb{B}\mathsf{P})_x$. We must show that $y := \mathrm{cod}(\tau \circ f_x)$ is sent by f to $z := \mathrm{cod}(\tau)$. To this end, consider the diagram to the left in:





The front and back faces are instances of (32), the left face commutes since f is a comonad morphism, and the bottom face commutes by naturality. We can thus read off that the outside of the diagram to the right commutes; as such, its upper composite (clearly) factors through ι_z, but

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

also through ι_{fy}, since $f \circ \iota_y = \iota_{fy} \circ f_y$: whence by Lemma 6.2(ii) we have $z = f(y)$, giving the first cofunctor axiom.

Finally, we address cofunctor axiom (iii). Note that we can now insert f_y into the diagram right above; whereupon the right square commutes by definition of f_y, and the left square since it does so on postcomposition by the monic ι_z. Postcomposing this left-hand square with some $\sigma : Q_z \to \mathrm{id}$ yields the final cofunctor axiom. □

Proposition 6.7 *For each morphism $f : \mathsf{P} \to \mathsf{Q}$ of accessible comonads, we have a commuting square of monad morphisms:*

$$\begin{array}{ccc} \mathsf{Q}^\circ & \xrightarrow{\theta} & \mathsf{T}_{\mathbb{B}_\mathsf{Q}} \\ {\scriptstyle f^\circ}\downarrow & & \downarrow{\scriptstyle \mathsf{T}_{\mathbb{B}_f}} \\ \mathsf{P}^\circ & \xrightarrow{\theta} & \mathsf{T}_{\mathbb{B}_\mathsf{P}} . \end{array} \tag{36}$$

Proof. For each $\tau : Q \to A \cdot \mathrm{id}$ in $Q^\circ A$, and each $x \in P1$, we have a diagram

$$\begin{array}{ccccc} P_x & \xrightarrow{f_x} & Q_{fx} & \xrightarrow{\tau_{fx}} & \mathrm{id} \\ {\scriptstyle \iota_x}\downarrow & & \downarrow{\scriptstyle \iota_{fx}} & & \downarrow{\scriptstyle \nu_{a_{fx}}} \\ P & \xrightarrow{f} & Q & \xrightarrow{\tau} & A \cdot \mathrm{id} \end{array}$$

whose right square is as in Proposition 6.4, and whose left square is the definition of f_x. It thus follows that the image of τ under $(T_{\mathbb{B}_f})_A \circ \theta^{\mathsf{Q}}_A$ is $\lambda x.\, (a_{fx}, \tau_{fx} \circ f_x)$. On the other hand, by unicity in Proposition 6.4, the image of $f^\circ(\tau) = \tau \circ f \in P^\circ A$ under $\theta^{\mathsf{P}}_A : P^\circ A \to T_{\mathbb{B}_\mathsf{P}} A$ is also $\lambda x.\, (a_{fx}, \tau_{fx} \circ f_x)$, as desired. □

Combining this result with Proposition 4.13, we obtain:

Theorem 6.8 *The functor $\mathbb{B}_{(-)} : \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et}) \to \mathcal{C}\mathrm{of}$ taking an accessible comonad Q to its behaviour category $\mathbb{B}_\mathsf{Q}$, and a map of accessible comonads $f : \mathsf{P} \to \mathsf{Q}$ to the cofunctor $\mathbb{B}_f : \mathbb{B}_\mathsf{P} \rightsquigarrow \mathbb{B}_\mathsf{Q}$ of Proposition 6.7, yields a within-isomorphism factorisation*

$$\begin{array}{ccc} & & \mathcal{C}\mathrm{of} \\ & {\scriptstyle \mathbb{B}_{(-)}} \nearrow \cong & \downarrow{\scriptstyle \mathsf{Q}_{(-)}} \\ \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et}) & \xrightarrow[\mathrm{Costr}]{} & \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}} . \end{array}$$

7. Idempotency of Costructure–Cosemantics

So far, we have seen that cosemantics takes values in presheaf comonads, and costructure takes values in presheaf monads; to complete our understanding of the costructure–cosemantics adjunction, we now show that further application of either adjoint simply interchanges a presheaf monad with its corresponding presheaf comonad. More precisely, we will show that the costructure–cosemantics adjunction is *idempotent*, with the presheaf monads and comonads as fixpoints to either side.

7.1. Idempotent adjunctions

We begin by recalling standard category-theoretic background on fixpoints and idempotency for adjunctions. To motivate this, recall that any adjunction between posets induces an isomorphism

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

between the sub-posets of fixpoints to each side. Similarly, any adjunction of categories restricts to an adjoint equivalence between the full subcategories of *fixpoints* in the following sense:

Definition 7.1 (Fixpoints). *Let $L \dashv R \colon \mathcal{D} \to \mathcal{C}$ be an adjunction. A* fixpoint *to the left is an object $X \in \mathcal{D}$ at which the counit map $\varepsilon_X \colon LRX \to X$ is invertible; a* fixpoint to the right *is $Y \in \mathcal{C}$ for which $\eta_Y \colon Y \to RLY$ is invertible. We write $\mathcal{F}\mathrm{ix}(LR)$ and $\mathcal{F}\mathrm{ix}(RL)$ for the full subcategories of fixpoints to the left and right.*

In the posetal case, the fixpoints to the left and the right are, respectively, coreflective and reflective in the whole poset. This is not true in general for adjunctions between categories, but it is true in the following situation:

Definition 7.2 (Idempotent adjunction). *An adjunction $L \dashv R \colon \mathcal{D} \to \mathcal{C}$ is called* idempotent *if it satisfies any one of the following equivalent conditions:*

(i) Each RX is a fixpoint;
(ii) R inverts each counit component;
(iii) The monad RL is idempotent;
(iv) Each LY is a fixpoint;
(v) L inverts each unit component;
(vi) The comonad LR is idempotent.

The equivalence of these conditions is straightforward and well known; for the reader who has not seen it, we leave the proof as an instructive exercise. Equally straightforward are the following consequences of the definition:

Proposition 7.3 *If the adjunction $L \dashv R \colon \mathcal{D} \to \mathcal{C}$ is idempotent, then:*

(i) $X \in \mathcal{D}$ is a fixpoint if and only if it is in the essential image of L;
(ii) $Y \in \mathcal{C}$ is a fixpoint if and only if it is in the essential image of R;
(iii) The fixpoints to the left are coreflective in $\mathcal{D}$ via $X \mapsto LRX$.
(iv) The fixpoints to the right are reflective in $\mathcal{C}$ via $Y \mapsto RLY$;

7.2. Presheaf monads and presheaf comonads are fixpoints

We aim to show that the costructure–cosemantics adjunction (21) is idempotent, with the presheaf monads and comonads as the fixpoints. We first show that costructure and cosemantics interchange a presheaf monad with the corresponding presheaf comonad.

Proposition 7.4 *We have isomorphisms of comonads, natural in $\mathbb{B}$, of the form*

$$\alpha_{\mathbb{B}} \colon \mathsf{Q}_{\mathbb{B}} \to \mathrm{Cosem}(\mathsf{T}_{\mathbb{B}}) \tag{37}$$

characterised by the fact that they induce on categories of Eilenberg–Moore coalgebras the functor $\mathbb{B}\text{-}\mathcal{S}\mathrm{et} \to {}^{\mathsf{T}_{\mathbb{B}}}\mathcal{S}\mathrm{et}$ sending the left $\mathbb{B}$-set $(X, p, *)$ to the $\mathsf{T}_{\mathbb{B}}$-comodel $\boldsymbol{X}$ with

$$\begin{aligned} [\![\lambda b.\,(f_b, a_b)]\!]_{\boldsymbol{X}} \colon X &\to A \times X \\ x &\mapsto (a_{p(x)}, f_{p(x)} * x)\,. \end{aligned} \tag{38}$$

In the statement of this result, we identify $\mathcal{C}\mathrm{oalg}(\mathsf{Q}_{\mathbb{B}})$ with $\mathbb{B}\text{-}\mathcal{S}\mathrm{et}$ by Proposition 3.10, and $\mathcal{C}\mathrm{oalg}(\mathrm{Cosem}(\mathsf{T}_{\mathbb{B}}))$ with ${}^{\mathsf{T}_{\mathbb{B}}}\mathcal{S}\mathrm{et}$ by Proposition 4.3.

Proof. By Proposition 3.5, the associated monad of the theory of $\mathbb{B}$-valued dependently typed update is the presheaf monad $\mathsf{T}_{\mathbb{B}}$; so by Proposition 2.28, we have an isomorphism over $\mathcal{S}\mathrm{et}$ of categories of comodels ${}^{\mathbb{T}_{\mathbb{B}}}\mathcal{S}\mathrm{et} \cong {}^{\mathsf{T}_{\mathbb{B}}}\mathcal{S}\mathrm{et}$, sending the $\mathbb{T}_{\mathbb{B}}$-comodel $\boldsymbol{X}$ to the $\mathsf{T}_{\mathbb{B}}$-comodel structure

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

on X with $[\![\lambda b.\,(f_b, a_b)]\!] = [\![\mathsf{get}(\lambda b.\,\mathsf{upd}_{f_b}(a_b))]\!]$. Composing this isomorphism with the invertible (20) yields an invertible functor $\mathbb{B}\text{-}\mathcal{S}\mathrm{et} \to {}^{\mathsf{T}_\mathbb{B}}\mathcal{S}\mathrm{et}$ over $\mathcal{S}\mathrm{et}$, which by inspection has the formula (38). We conclude by the full fidelity of the Eilenberg–Moore semantics functor (Lemma 4.2). □

Proposition 7.5 *For any small category $\mathbb{B}$, the monad morphism*

$$\bar{\alpha}_\mathbb{B} : \mathsf{T}_\mathbb{B} \to \mathrm{Costr}(\mathsf{Q}_\mathbb{B}) \tag{39}$$

found as the adjoint transpose of the isomorphism (37), is itself an isomorphism.

Proof. By Remark 4.10 and (38), we see that $\bar{\alpha}$ sends the element $(f, a) = \lambda b.\,(f_b, a_b)$ of $T_\mathbb{B}(A) = \prod_b (\mathbb{B}_b \times A)$ to the transformation $\bar{\alpha}(f,a) : U^\mathbb{B} \Rightarrow A \cdot U^\mathbb{B} : \mathbb{B}\text{-}\mathcal{S}\mathrm{et} \to \mathcal{S}\mathrm{et}$ whose component at a $\mathbb{B}$-set $(X, p, \cdot)$ is given by the function

$$\begin{aligned} \bar{\alpha}(f,a)_{(X,p,\cdot)} : X &\to A \times X \\ x &\mapsto (a_{p(x)}, f_{p(x)} \cdot x)\,. \end{aligned} \tag{40}$$

We must show every $\gamma : U^\mathbb{B} \Rightarrow A \cdot U^\mathbb{B}$ takes this form for a unique $(f, a) \in T_\mathbb{B}(A)$. For each $b \in \mathbb{B}$, we have the representable left $\mathbb{B}$-set $y(b)$ with underlying set $\mathbb{B}_b$, projection to $\mathrm{ob}(\mathbb{B})$ given by codomain, and action given by composition in $\mathbb{B}$. The component of γ at $y(b)$ is a function $\mathbb{B}_b \to A \times \mathbb{B}_b$, which, if we are to have $\gamma = \bar{\alpha}(f,a)$, must by (40) have its value at $1_b \in \mathbb{B}_b$ given by (a_b, f_b). Thus, if we define (a_b, f_b) to be $\gamma_{y(b)}(1_b)$ for each $b \in \mathbb{B}$, then it remains only to verify that indeed $\gamma = \bar{\alpha}(f,a)$. But for any $\mathbb{B}$-set $(X, p, *)$ and any $x \in X$, the Yoneda lemma states that there is a unique map of $\mathbb{B}$-sets $y(p(x)) \to X$ sending 1_{px} to x, namely $\tilde{x}$ given by $\tilde{x}(f) \mapsto f * x$. Now naturality of γ ensures that

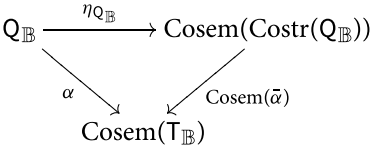
$$\gamma_{(X,p,*)}(x) = \gamma_{(X,p,*)}(\tilde{x}(1_{px})) = (A \times \tilde{x})(\gamma_{y(px)}(1_{px})) = (a_b, \tilde{x}(f_{px})) = (a_{px}, f_{px} * x)$$

so that $\gamma = \bar{\alpha}(f,a)$ as desired. □

Given the tight relationship between (37) and (39), it is now easy to conclude that presheaf monads and comonads are fixpoints.

Proposition 7.6 *Each presheaf monad is a fixpoint to the left of the costructure–cosemantics adjunction, while each presheaf comonad is a fixpoint to the right.*

Proof. For each small category $\mathbb{B}$, we have $\bar{\alpha} : \mathsf{T}_\mathbb{B} \to \mathrm{Costr}(\mathsf{Q}_\mathbb{B})$ as in (37) and its adjoint transpose $\alpha : \mathsf{Q}_\mathbb{B} \to \mathrm{Cosem}(\mathsf{T}_\mathbb{B})$ as in (39). By definition of adjoint transpose, this $\bar{\alpha}$ is the unique map fitting into a commuting triangle



where $\eta_{\mathsf{Q}_\mathbb{B}}$ is the unit of (21). Since both α and $\bar{\alpha}$ are invertible, it follows that $\eta_{\mathsf{Q}_\mathbb{B}}$ is too, and since every presheaf comonad is isomorphic to some $\mathsf{Q}_\mathbb{B}$, it follows that every presheaf comonad is a fixpoint on the right. The dual argument shows each presheaf monad is a fixpoint on the left. □

7.3. Idempotency of the costructure–cosemantics adjunction

As an immediate consequence of the preceding result, we have:

Theorem 7.7 *The costructure–cosemantics adjunction (21) is idempotent. Its fixpoints to the left are the presheaf monads, while those to the right are the presheaf comonads.*

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

Proof. Each Cosem(T) is a presheaf comonad by Proposition 5.1, and each presheaf comonad is a fixpoint to the right by Proposition 7.6; thus, Definition 7.2(i) is satisfied and the adjunction is idempotent. For the remaining claims, one direction is Proposition 7.6, while the other follows on noting that, by the preceding result and Proposition 7.5, the composite Cosem ∘ Costr sends each comonad to a presheaf comonad, while Costr ∘ Cosem sends each monad to a presheaf monad. □

We may use this result to resolve some unfinished business:

Proposition 7.8 *The presheaf monad functor* $\mathsf{T}_{(-)} \colon \mathcal{C}\mathrm{of} \to \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}}$ *of Proposition 3.8 is full and faithful.*

Proof. Since the costructure–cosemantics adjunction is idempotent, the functor $\mathrm{Costr} \colon \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et}) \to \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}}$ is fully faithful when restricted to the subcategory of presheaf comonads; and since $\mathsf{Q}_{(-)}$ takes its image in this subcategory, we see that $\mathrm{Costr} \circ \mathsf{Q}_{(-)} \colon \mathcal{C}\mathrm{of} \to \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}}$ is fully faithful. Now transporting the values of this composite functor along the isomorphisms $\bar{\alpha}_{\mathbb{B}} \colon \mathsf{T}_{\mathbb{B}} \cong \mathrm{Costr}(\mathsf{Q}_{\mathbb{B}})$ of Proposition 7.5 yields a fully faithful functor $\mathcal{C}\mathrm{of} \to \mathrm{Mnd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}}$ which acts on objects by $\mathbb{B} \mapsto \mathsf{T}_{\mathbb{B}}$, and on morphisms by $F \mapsto (\bar{\alpha}_{\mathbb{B}})^{-1} \circ \mathrm{Costr}(\mathsf{Q}_F) \circ \bar{\alpha}_{\mathbb{C}}$. Now direct calculation shows this action on morphisms to be precisely that of (28). □

It follows from this and Proposition 3.7 that the full embeddings $\mathsf{Q}_{(-)} \colon \mathcal{C}\mathrm{of} \to \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et})$ and $\mathsf{T}_{(-)} \colon \mathcal{C}\mathrm{of} \to \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}}$ exhibit Cof as equivalent to the full subcategories of fixpoints to the left and to the right; from which it follows that:

Proposition 7.9 *The presheaf monad and presheaf comonad functors of Propositions 3.8 and 3.7, together with the behaviour functors of Theorems 5.23 and 6.8, participate in adjunctions*

$$\mathcal{C}\mathrm{of} \underset{\mathsf{T}_{(-)}}{\overset{\mathbb{B}_{(-)}}{\underset{\top}{\leftarrow}}} \mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})^{\mathrm{op}} \qquad \mathcal{C}\mathrm{of} \underset{\mathsf{Q}_{(-)}}{\overset{\mathbb{B}_{(-)}}{\underset{\bot}{\leftarrow}}} \mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et})$$

exhibiting the full subcategories of presheaf monads, respectively, presheaf comonads, as reflective in $\mathcal{M}\mathrm{nd}_a(\mathcal{S}\mathrm{et})$*, respectively,* $\mathcal{C}\mathrm{md}_a(\mathcal{S}\mathrm{et})$.

We can describe the units of these reflections explicitly. On the one hand, if Q is an accessible comonad on $\mathcal{S}\mathrm{et}$, then its reflection in the full subcategory of presheaf comonads is the presheaf comonad of the behaviour category $\mathbb{B}_{\mathsf{Q}}$, and the reflection map $\mathsf{Q} \to \mathsf{Q}_{\mathbb{B}_{\mathsf{Q}}}$ has components

$$\eta_A \colon Q(A) \to \textstyle\sum_{x \in Q1} A^{[\mathcal{S}\mathrm{et},\mathcal{S}\mathrm{et}](Q_x,\mathrm{id})}$$
$$a \mapsto (Q!(a), \lambda\tau.\, \tau_A(a))\,.$$

On the other hand, if T is an accessible monad on $\mathcal{S}\mathrm{et}$, then its reflection into the full subcategory of presheaf monads is the presheaf monad of the behaviour category $\mathbb{B}_{\mathsf{T}}$, and the reflection map $\eta \colon \mathsf{T} \to \mathsf{T}_{\mathbb{B}_{\mathsf{T}}}$ has components

$$\eta_A \colon T(A) \to \textstyle\prod_{\beta \in B_{\mathsf{T}}} (T(1)/{\sim_\beta} \times A)$$
$$t \mapsto \lambda\beta.\, (\tilde{t}, \beta(t))\,.$$

In fact, this reflection map exhibits $\mathsf{T}_{\mathbb{B}_{\mathsf{T}}}$ as the result of adjoining to T a new B_{T}-ary operation beh satisfying the axioms of read-only state and the axioms

$$t(u) \equiv_{\mathsf{beh},\beta} t \gg u_{\beta(t)}$$

for all $t \in T(A)$ and $u \in T(B)^A$. From a computational perspective, we understand the new operation beh as an "oracle" that allows the user to request complete information about the future behaviour of the external system with which we are interacting. Of course, since this future

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

behaviour is rarely computable in terms of the generating operations of T, we immediately leave the realm of computationally meaningful theories. In future work, we will see how to rectify this, to some degree, by considering an adjunction between accessible monads on $\mathcal{S}\mathrm{et}$ and suitably accessible comonads on the category of *topological spaces*. In this refined setting, we will see that the passage from monad to comonad and back adjoins new operations which observe (via the primitives of $\mathbb{T}$) only *finite* amounts of information about the future behaviour of the system.

8. Examples and Applications: Cosemantics

In the final two sections of this paper, we give a range of examples illustrating our main results. In this section, we calculate the behaviour category, and the comodels classifying admissible behaviours, for a range of examples of algebraic theories for computational effects, and calculate some examples of cofunctors between behaviour categories induced by computationally interesting interpretations of algebraic theories.

8.1. Reversible input

Given a set V, the theory of *V-valued reversible input* (first considered for $|V| = 2$ in [18]) is generated by a V-ary operation read, and a V-indexed family of unary operations unread_v, satisfying the equations

$$\mathsf{unread}_v(\mathsf{read}(x)) \equiv x_v \qquad \text{and} \qquad \mathsf{read}(\lambda u.\, \mathsf{unread}_u(x)) \equiv x\,. \tag{41}$$

If read is thought of as reading the next value from an input stream, then unread_v returns the value v to the front of that stream. A comodel of this theory comprises the data of a set S, a function $[\![\mathsf{read}]\!] = (g, n)\colon S \to V \times S$ and functions $[\![\mathsf{unread}_v]\!]\colon S \to S$, or equally a single function $p\colon V \times S \to S$; while the equations force (g,n) and p to be inverse to each other. Thus, comodels of V-valued reversible input are equally well comodels of V-valued input whose structure map $[\![\mathsf{read}]\!]\colon S \to V \times S$ is invertible. Since, in particular, this is true for the final comodel $\boldsymbol{V}^{\mathbb{N}}$ of V-valued input by the well-known Lambek lemma, we conclude that this is also the final comodel of V-valued reversible input.

We now calculate the comodel associated with an admissible behaviour $W \in V^{\mathbb{N}}$. We begin with some calculations relating to $\sim_W$-equivalence. First, by Remark 5.12, any unary term is $\sim_W$-equivalent to one of the form $\sigma_1 \gg \cdots \gg \sigma_n \gg \mathrm{id}$ where each σ_i is either read or some unread_v. Now the first equation in (41) implies that $\mathsf{unread}_v \gg \mathsf{read} \gg (\text{–})$ is the identity operator, and so any unary term is $\sim_W$-equivalent to one of the form

$$[n, v_m, \ldots, v_1] := \overbrace{\mathsf{read} \gg \cdots \gg \mathsf{read}}^{n} \gg \mathsf{unread}_{v_m} \gg \cdots \gg \mathsf{unread}_{v_1} \gg \mathrm{id}$$

for some $n \in \mathbb{N}$ and $v_m, \ldots, v_1 \in V$. Since the behaviour W satisfies $W(\mathsf{read}) = W_0$, we have $\mathsf{read} \gg \mathsf{unread}_{W_0} = \mathsf{read}(\lambda u.\, \mathsf{unread}_{W_0}) \sim_W \mathsf{read}(\lambda u.\, \mathsf{unread}_u) = \mathrm{id}$; whence by Lemma 5.14, also $[n+1, W_n, v_m, \ldots, v_1] \sim_W [n, v_m, \ldots, v_1]$ for any $n \in \mathbb{N}$ and $v_m, \ldots, v_1 \in V$. Consequently, each unary term is $\sim_W$-equivalent to an element of the set S_W given by

$$\{[n, v_m, \ldots, v_1] : n, m \in \mathbb{N}, v_i \in V \text{ and } W_{n-1} \neq v_m \text{ if } n, m > 0\}\,. \tag{42}$$

We claim that S_W is in fact a set of $\sim_W$-equivalence class representatives. For this, it suffices by Remark 5.19 to endow S_W with a comodel structure $\boldsymbol{S}_W$ satisfying (31) – which will then make it the comodel classifying states of behaviour W. We do so by taking $[\![\mathsf{read}]\!]_{\boldsymbol{S}_W}$ to be given by

$$[n, v_m, \ldots, v_1] \mapsto \begin{cases} (W_n, [n+1]) & \text{if } m = 0; \\ (v_1, [n, v_m, \ldots, v_2]) & \text{if } m > 0, \end{cases} \tag{43}$$

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

and taking $[\![\mathsf{unread}_v]\!]_{S_W}$ to be given by

$$[n, v_1, \ldots, v_m] \mapsto \begin{cases} [n-1] & \text{if } m = 0,\ n > 0,\ v = W_{n-1}; \\ [n, v_m, \ldots, v_1, v] & \text{otherwise.} \end{cases} \tag{44}$$

We may now use the above calculations to identify morphisms $W \to W'$ in the behaviour category. These correspond to comodel homomorphisms $\boldsymbol{S}_{W'} \to \boldsymbol{S}_W$ and so to states of $\boldsymbol{S}_W$ of behaviour W'. Since the state $[n, v_m, \ldots, v_1] \in \boldsymbol{S}_W$ has behaviour given by the stream of values $v_1 \ldots v_m W_n W_{n+1} W_{n+2} \ldots$, we conclude that morphisms $W \to W'$ in the behaviour category are states of the form $[n, {W'}_{m-1}, \ldots, {W'}_0]$ where ${W'}_k = W_{k+n-m}$ for all $k \geqslant m$ but ${W'}_{m-1} \neq W_{n-1}$. Such a state is clearly uniquely determined by the integer $i = n - m$, and so we arrive at:

Proposition 8.1 *The behaviour category of the theory of V-valued reversible input has object-set $V^{\mathbb{N}}$; morphisms $W \to W'$ are integers i such that, for some $N \in \mathbb{N}$, we have ${W'}_k = W_{k+i}$ for all $k > N$; and composition is addition of integers. The comodel classifying states of behaviour $W \in V^{\mathbb{N}}$ has underlying set (42), and co-operations as in (43) and (44).*

Note that this behaviour category is a groupoid; in fact, it is not hard to show that it is the *free* groupoid on the behaviour category for V-valued input. This groupoid is well known in the study of Cuntz C^*-algebras: for example, for finite V it appear already in [33, Definition III.2.1]. In this context, it is important that the groupoid is not just as a groupoid of sets, but a *topological* groupoid; in a sequel to this paper, we will explain how this topology arises very naturally via comodels.

8.2. Stack

Given a set V, the theory of a *V-valued stack* – introduced for a finite V in [13] – is generated by a $V + \{\bot\}$-ary operation pop, whose arguments we group into an V-ary part and a unary part; and a V-indexed family of unary operations push_v for $v \in V$, satisfying the equations

$$\mathsf{push}_v(\mathsf{pop}(x, y)) \equiv x_v \quad \mathsf{pop}(\lambda v.\, \mathsf{push}_v(x), x) \equiv x \quad \mathsf{pop}(x, \mathsf{pop}(y, z)) \equiv \mathsf{pop}(x, z)\,.$$

This theory captures the semantics of a stack of elements from V: we read $\mathsf{push}_v(x)$ as "push v on the stack and continue as x", and $\mathsf{pop}(x, y)$ as "if the stack is non-empty, pop its top element v and continue as x_v; else continue as y".

Note the similarities with the theory of V-valued reversible input; indeed, this latter theory could equally well be seen as the theory of a V-valued *infinite* stack. We can formalise this via an interpretation of the theory of V-valued stack into V-valued reversible input which maps push_u to unread_u and $\mathsf{pop}(x, y)$ to $\mathsf{read}(x)$.

A comodel of the theory of a V-valued stack comprises a set S with functions $(g, n)\colon S \to (V + \{\bot\}) \times S$ (modelling pop) and $p\colon V \times S \to S$ (modelling the push_v's) subject to conditions corresponding to the three equations above:

(1) $g(p(v, s)) = v$ and $n(p(v, s)) = s$;
(2) If $g(s) = \bot$ then $n(s) = s$, while if $g(s) = v$ then $p(v, n(s)) = s$;
(3) If $g(s) = \bot$ then $g(n(s)) = \{\bot\}$ and $n(n(s)) = n(s)$ (this is implied by (2)).

Writing $E = \{s \in S : g(s) = \bot\}$ for the set of "states in which the stack is empty", and $j\colon E \to S$ for the inclusion, (1) implies that $p\colon V \times S \to S$ is an injection whose image is disjoint from that

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

of j and (2) that every $s \in S$ lies either in E or in the image of p. So we have a coproduct diagram

$$\begin{array}{ccccc} V \times S & & & & E\,. \\ & {\scriptstyle p}\searrow & & \swarrow{\scriptstyle j} & \\ & & S & & \end{array}$$

In fact, any such coproduct diagram comes from a comodel: for indeed, we may recover the morphism $(g, n)\colon S \to (V + \{\bot\}) \times S$ as the unique map whose composites with p and j are $\lambda(v, s).\,(v, s)$ and $\lambda e.\,(\bot, j(e))$, respectively. Thus, a comodel structure on a set S is equivalently given by a set E and a coproduct diagram $V \times S \to S \leftarrow E$.

The *final* comodel of this theory is the set $V^{\leqslant\omega}$ of partial functions $\mathbb{N} \rightharpoonup V$ which are defined on some initial segment of $\mathbb{N}$, under the comodel structure corresponding to the coproduct diagram

$$\begin{array}{ccccc} V \times V^{\leqslant\omega} & & & & \{*\}\,. \\ & {\scriptstyle (v,W)\mapsto v.W}\searrow & & \swarrow{\scriptstyle *\mapsto\varepsilon} & \\ & & S & & \end{array}$$

Here, we write ε for the everywhere-undefined element of $V^{\leqslant\omega}$, and write $v.W \in V^{\leqslant\omega}$ for the element with $(v.W)_0 = v$ and $(v.W)_{i+1} \simeq W_i$[2]. In terms of the generating co-operations, this final comodel is given by

$$[\![\mathsf{push}_v]\!](W) = v.W \qquad [\![\mathsf{pop}]\!](v.W) = (v, W) \qquad \text{and} \qquad [\![\mathsf{pop}]\!](\varepsilon) = (\bot, \varepsilon)\,.$$

We now calculate the comodel associated with an admissible behaviour $W \in V^{\leqslant\omega}$. Given the similarity with the theory of V-valued reversible input, we may argue as in the previous section to see that any unary term is $\sim_W$-equivalent to one

$$[n, v_m, \ldots, v_1] := \underbrace{\mathsf{pop} \gg \cdots \gg \mathsf{pop}}_{n} \gg \mathsf{push}_{v_m} \gg \cdots \gg \mathsf{push}_{v_1} \gg \mathsf{id}$$

for some $n \in \mathbb{N}$ and $v_m, \ldots, v_1 \in V$. Now, if W_0 is undefined, then $W(\mathsf{pop}) = \bot$, and so $\mathsf{pop} \gg m = \mathsf{pop}(\lambda v.\, m, m) \sim_W \mathsf{pop}(\lambda v.\, \mathsf{push}_v(m), m) = m$ for any $m \in T(1)$. By Lemma 5.14, it follows that $[n+1, v_m, \ldots, v_1] \sim_W [n, v_m, \ldots, v_1]$ whenever W_n is undefined, and so we conclude that each unary term is $\sim_W$-equivalent to some $[n, v_m, \ldots, v_1]$ for which W is defined at all $k < n$. At this point, by repeating the arguments of the preceding section, *mutatis mutandis*, we may show that any unary term is $\sim_W$-equivalent to an element of the set

$$\{[n, v_m, \ldots, v_1] : n, m \in \mathbb{N}, v_i \in V, W \text{ defined at all } k < n, \text{ and } W_{n-1} \neq v_m \text{ if } n, m > 0\}\,. \quad (45)$$

We now show, like before, that this is a set of $\sim_W$-equivalence class representatives, by making it into a comodel satisfying (31); again, this comodel will then classify states of behaviour W. This time, we take $[\![\mathsf{pop}]\!]$ to be given by

$$[n, v_m, \ldots, v_1] \mapsto \begin{cases} (W_n, [n+1]) & \text{if } m = 0 \text{ and } W_n \text{ defined;} \\ (\bot, [n]) & \text{if } m = 0 \text{ and } W_n \text{ undefined;} \\ (v_1, [n, v_m, \ldots, v_2]) & \text{if } m > 0, \end{cases} \quad (46)$$

and take $[\![\mathsf{push}_v]\!]$ to be given exactly as in (44). Transcribing the calculations of the preceding section, we arrive at:

Proposition 8.2 *The behaviour category of the theory of a V-valued stack has object-set $V^{\leqslant\omega}$; morphisms $W \to W'$ are integers i such that, for some $N \in \mathbb{N}$, we have ${W'}_k \simeq W_{k+i}$ for all $k > N$,*

[2] We use *Kleene equality* $a \simeq b$, meaning that a is defined just when b is defined, and they are then equal.

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

and composition is addition of integers. The comodel classifying states of behaviour $W \in V^{\leqslant\omega}$ has underlying set (45), and co-operations as in (46) and (44).

In fact, it is easy to see that the behaviour category of a V-valued stack is the disjoint union of the behaviour category for V^*-valued state (modelling a finite stack) and for V-valued reversible input (modelling an infinite stack). The cofunctor on behaviour categories induced by the interpretation of the theory of a V-valued stack into that of V-valued reversible input is simply the connected component inclusion.

8.3. Dyck words

A *Dyck word* is a finite list $W \in \{\mathsf{U}, \mathsf{D}\}^*$ with the same number of U's as D's, and with the property that the ith U in the list always precedes the ith D. Here, U and D stand for "up" and "down", and the idea is that a Dyck word records a walk on the natural numbers $\mathbb{N}$ with steps ± 1 which starts and ends at 0. More generally, we can encode walks from $n \in \mathbb{N}$ to $m \in \mathbb{N}$ by "affine Dyck words":

Definition 8.3 (Affine Dyck words). *Given $n, m \in \mathbb{N}$, an* affine Dyck word *from n to m is a word $W \in \{\mathsf{U}, \mathsf{D}\}^*$ such that $\#\{\mathsf{D}\text{'s in } W\} - \#\{\mathsf{U}\text{'s in } W\} = n - m$, and such that the ith U precedes the $(i+n)$th D for all suitable i. We may extend this notation by declaring* any *word $W \in \{\mathsf{U}, \mathsf{D}\}^*$ to be an affine Dyck word from ∞ to ∞. If $n, m \in \mathbb{N} \cup \{\infty\}$, then we write $W : n \rightsquigarrow m$ to indicate that W is an affine Dyck word from n to m.*

For example:

- UUDUDD is a Dyck word, but also an affine Dyck word $n \rightsquigarrow n$ for any n;
- UDDUUU is an affine Dyck word $1 \rightsquigarrow 3$ and $2 \rightsquigarrow 4$, but not $0 \rightsquigarrow 2$.

We now describe an algebraic theory which encodes the dynamics of the walks encoded by affine Dyck words. It has two unary operations U and D; and an $\mathbb{N}$-indexed family of binary operations $\mathsf{ht}_{>n}$ each satisfying the axioms of read-only state; all subject to the following axioms:

$$\mathsf{ht}_{>n}(x, \mathsf{ht}_{>m}(y, z)) \equiv \mathsf{ht}_{>m}(\mathsf{ht}_{>n}(x, y), z)$$
$$\mathsf{ht}_{>0}(x, \mathsf{D}(x)) \equiv x \qquad \mathsf{U}(\mathsf{ht}_{>0}(x, y)) \equiv \mathsf{U}(x)$$
$$\mathsf{U}(\mathsf{ht}_{>n+1}(x, y)) \equiv \mathsf{ht}_{>n}(\mathsf{U}(x), \mathsf{U}(y)) \qquad \mathsf{D}(\mathsf{ht}_{>n}(x, y)) \equiv \mathsf{ht}_{>n+1}(\mathsf{D}(x), \mathsf{D}(y))$$

for all $m \leqslant n \in \mathbb{N}$. The theory of affine Dyck words provides an interface for accessing a state machine with an internal "height" variable $h \in \mathbb{N}$, which responds to two commands U and D which, respectively, increase and decrease h by one, with the proviso that D should do nothing when applied in a state with $h = 0$. With this understanding, we read the primitive $\mathsf{ht}_{>n}(x, y)$ as "if $h > n$ then continue as x, else continue as y"; read $\mathsf{U}(x)$ as "perform U and continue as x"; and read $\mathsf{D}(x)$ as "perform D (so long as $h > 0$) and continue as x".

Rather than compute the comodels by hand, we pass directly to a calculation of the behaviour category. We begin by finding the admissible behaviours. By Remark 5.6 and the fact that $\mathsf{ht}_{>n} \gg (-)$ is the identity operator, an admissible behaviour β is uniquely determined by the values $\beta(\sigma_1 \gg \cdots \gg \sigma_k \gg \mathsf{ht}_{>n})$ where each σ_i is either U or D. Now, the last three axioms imply that these values are determined in turn by the values $\beta(\mathsf{ht}_{>n}) \in \{\mathsf{tt}, \mathsf{ff}\}$ for each n. Finally, by the first axiom, $\beta(\mathsf{ht}_{>m}) = \mathsf{ff}$ implies $\beta(\mathsf{ht}_{>n}) = \mathsf{ff}$ whenever $m \leqslant n$. So the possibilities are either that there is a *least* n with $\beta(\mathsf{ht}_{>n}) = \mathsf{ff}$, or that $\beta(\mathsf{ht}_{>n}) = \mathsf{tt}$ for all $n \in \mathbb{N}$.

In fact, each of these possibilities for β does yields an admissible behaviour. Indeed, identifying these possibilities with elements of the set $\mathbb{N} \cup \{\infty\}$, we can try to make this set into a comodel via

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

the formulae of Proposition 5.9, by taking:

$$[\![\mathsf{ht}_{>n}]\!](k) = \begin{cases} (\mathsf{tt}, k) & \text{if } k > n; \\ (\mathsf{ff}, k) & \text{otherwise,} \end{cases} \qquad [\![\mathsf{U}]\!](k) = k+1\,, \quad [\![\mathsf{D}]\!](k) = \begin{cases} 0 & \text{if } k = 0; \\ n-1 & \text{otherwise,} \end{cases}$$

where we take $\infty > n$ for any $n \in \mathbb{N}$, and $\infty + 1 = \infty = \infty - 1$. It is not hard to check that these co-operations do in fact yield a comodel, which is then of necessity the final comodel of the theory of affine Dyck words.

We now compute the comodel $\boldsymbol{k}$ associated with a behaviour $k \in \mathbb{N} \cup \{\infty\}$. Because each operator $\mathsf{ht}_{>n} \gg (-)$ is the identity, each unary operation is $\sim_k$-equivalent to one in the submonoid generated by $\mathsf{U}, \mathsf{D} \in T(1)$. Further, by the second axiom we have $\mathsf{D} \sim_0 \mathrm{id}$, and so by Lemma 5.14, also $W\mathsf{D}W' \sim_k WW'$ for any $k \in \mathbb{N}$, any $W' \in \{\mathsf{U}, \mathsf{D}\}^*$ and any affine Dyck word $W\colon k \rightsquigarrow 0$ from k to 0. Applying this rewrite rule repeatedly, we find that any unary term is $\sim_k$ equivalent to an element of the set

$$\{\, W \in \{\mathsf{U}, \mathsf{D}\}^* : W\colon k \rightsquigarrow \ell \text{ for some } \ell \in \mathbb{N} \cup \{\infty\} \,\}\,, \tag{47}$$

and we may apply Remark 5.19 to see that there are in fact no further relations. Indeed, we may make (47) into a classifying comodel $\boldsymbol{k}$ satisfying (31) by taking

$$\begin{gathered} [\![\mathsf{ht}_{>n}]\!](W) = \begin{cases} (\mathsf{tt}, W) & \text{if } W\colon k \rightsquigarrow \ell \text{ with } \ell > n; \\ (\mathsf{ff}, W) & \text{otherwise,} \end{cases} \\ [\![\mathsf{U}]\!](W) = W\mathsf{U}\,, \qquad [\![\mathsf{D}]\!](\mathsf{W}) = \begin{cases} W & \text{if } W\colon k \rightsquigarrow 0; \\ W\mathsf{D} & \text{otherwise.} \end{cases} \end{gathered} \tag{48}$$

From the preceding calculations, we can now read off:

Proposition 8.4 *The behaviour category of the theory of affine Dyck words has object-set* $\mathbb{N} \cup \{\infty\}$*, and morphisms from n to m given by affine Dyck words* $W\colon n \rightsquigarrow m$*. Composition is given by concatenation of words. The comodel classifying the behaviour* $k \in \mathbb{N} \cup \{\infty\}$ *has underlying set (47), and co-operations as in (48).*

The set of Dyck words of length $2n$ is well known to have the cardinality of the nth Catalan number $C_n = \frac{1}{n+1}\frac{2n}{n}$. On the other hand, C_n also enumerates the set of well-bracketed expressions, such as $((aa)a)(aa)$, composed of $n+1$ a's. In fact, there is a bijection between Dyck words of length $2n$ and well-bracketed expressions of $(n+1)$ a's which can be obtained by interpreting a Dyck word W as a set of instructions for a stack machine, as follows:

(1) Begin with a stack containing the single element a;
(2) Read the next element of the Dyck word W:
   - If it is U, push an a onto the stack;
   - If it is D, pop the top two elements x,y of the stack and push (xy) onto the stack.
(3) When W is consumed, return the single element remaining on the stack.

While the terms on which this stack machine operates are the well-bracketed expressions of a's, we can do something similar for stacks of elements of any set V endowed with a constant a and a binary operation $*$, obtaining for each Dyck word W an element of V built from a's and $*$'s. We

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

can understand this in terms of an interpretation $f\colon \mathbb{T}_{\mathrm{Dyck}} \to \mathbb{T}_{\mathrm{Stack}}$ of the theory of affine Dyck words into the theory of a V-valued stack, given as follows:

$$(\mathsf{ht}_{>0})^f(x,y) = \mathsf{pop}(\lambda v.\, \mathsf{push}_v(x), y)$$
$$(\mathsf{ht}_{>n+1})^f(x,y) = \mathsf{pop}(\lambda v.\, (\mathsf{ht}_{>n})^f(\mathsf{push}_v(x), \mathsf{push}_v(y)), y)$$
$$\mathsf{U}^f(x) = \mathsf{push}_a(x) \qquad \mathsf{D}^f(x) = \mathsf{pop}(\lambda v.\, \mathsf{pop}(\lambda w.\, \mathsf{push}_{v*w}(x), x), y)$$

Here, for the (recursively defined) interpretation of the predicates $\mathsf{ht}_{>n}$, we attempt to pop $n+1$ elements from the top of our stack of V's; if this succeeds, then we undo our pushes and return tt, while if it at any point fails, then we undo our pushes and return ff. For the interpretation of U we simply push our constant $a \in V$ onto the stack; while for the interpretation of $\mathsf{D}(x)$, we attempt to pop the top two elements v,w from the stack and push $v * w$ back on. If this succeeds, we continue as x, but some care is needed if it fails. By the fifth affine Dyck word equation, if our stack contains exactly one element, then D^f should yield a stack with no elements and continue as x; while by the second equation, if our stack is empty, then D^f should do nothing and continue as x. The forces the definition given above.

The *dynamics* of the interpretation of (affine) Dyck words as stack operations is captured by the induced cofunctor $\mathbb{B}_f\colon \mathbb{B}_{\mathrm{Stack}} \to \mathbb{B}_{\mathrm{Dyck}}$. It is an easy calculation to see that this is given as follows:

- On objects, we map $S \in V^{\leqslant\omega}$ to the cardinality $|S| \in \mathbb{N} \cup \{\infty\}$ of the initial segment of $\mathbb{N}$ on which S is defined.
- On morphisms, given $S \in V^{\leqslant\omega}$ and an affine Dyck word $W\colon |S| \rightsquigarrow k$, we return the morphism $S \to S'$ which updates the stack S via the sequence of U's and D's which specifies W.

In particular, we may consider the case where $(V, *, a)$ is the set of well-bracketed expressions of a's under concatenation. Now give a Dyck word W, we may regard it as an affine Dyck word $W\colon 1 \rightsquigarrow 1$; and now updating the singleton stack a via $W\colon 1 = |a| \rightsquigarrow 1$ yields precisely the well-bracketed expression of a's which corresponds to the given Dyck word W.

8.4. Store

Given a set L of *locations* and a family $\vec{V} = (V_\ell : \ell \in L)$ of *value* sets, the theory of $\vec{V}$*-valued store* comprises a copy of the theory of V_ℓ-valued state for each $\ell \in L$ – with operations $(\mathsf{put}_v^{(\ell)} : v \in V_\ell)$ and $\mathsf{get}^{(\ell)}$, say – together with, for all $\ell \neq k \in L$, all $v \in V_\ell$ and all $w \in V_k$, the commutativity axiom:

$$\mathsf{put}_v^{(\ell)}(\mathsf{put}_w^{(k)}(x)) \equiv \mathsf{put}_w^{(k)}(\mathsf{put}_v^{(\ell)}(x))\,. \tag{49}$$

By the argument of [15, Lemma 3.21 and 3.22], these equations also imply the commutativity conditions $\mathsf{put}_v^{(\ell)}(\mathsf{get}^{(k)}(\lambda w.\, x_w)) \equiv \mathsf{get}^{(k)}(\lambda w.\, \mathsf{put}_v^{(\ell)}(x_w))$ and $\mathsf{get}^{(\ell)}(\lambda v.\, \mathsf{get}^{(k)}(\lambda w.\, x_{vw})) \equiv \mathsf{get}^{(k)}(\lambda w.\, \mathsf{get}^{(\ell)}(\lambda v.\, x_{vw}))$.

A set-based comodel of this theory is a set S endowed with an L-indexed family of lens structures $(g^{(\ell)}\colon S \to V_\ell,\ p^{(\ell)}\colon V_\ell \times S \to S)$ which *commute* in the sense that

$$p^{(k)}(v, p^{(\ell)}(u,s)) = p^{(\ell)}(u, p^{(k)}(v,s))$$

for all $\ell, k \in L$, $u \in V_\ell$ and $v \in V_k$. When L is finite and each V_ℓ is the same set V, this is the notion of *array* given in [32, Section 4]. The final comodel of this theory is the set $\prod_{\ell \in L} V_\ell$, under the operations

$$[\![\mathsf{get}^{(\ell)}]\!](\vec{v}) = (v_\ell, \vec{v}) \qquad [\![\mathsf{put}_v^{(\ell)}]\!](\vec{v}) = \vec{v}[v/v_\ell]\,.$$

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

By similar arguments to those of the preceding sections, we may now show that:

Proposition 8.5 *The behaviour category $\mathbb{B}_{\vec{V}}$ of $\vec{V}$-valued store has set of objects $\prod_{\ell \in L} V_\ell$, while a morphism $\vec{v} \to \vec{w}$ is unique when it exists and exists just when $\vec{v}$ and $\vec{w}$ differ in only finitely many positions. The comodel classifying the behaviour $\vec{v} \in \prod_{\ell \in L} V_\ell$ is the sub-comodel of the final comodel on the set*

$$\{\vec{w} \in \textstyle\prod_{\ell \in L} V_\ell : \vec{v} \text{ and } \vec{w} \text{ differ in only finitely many positions}\}\,.$$

For each $\ell \in L$, there is an obvious interpretation of the theory of V_ℓ-valued state into the theory of $\vec{V}$-valued store, and this induces a cofunctor on behaviour categories $\mathbb{B}_{\vec{V}} \to \nabla V_\ell$ (recalling from Example 5.20 that ∇V_ℓ is the codiscrete category on V_ℓ) which:

- On objects, maps $\vec{v} \in \mathbb{B}_{\vec{V}}$ to its component $v_\ell \in \nabla V_\ell$;
- On morphisms, for each $\vec{v} \in \mathbb{B}_{\vec{V}}$, sends $v_\ell \to v'_\ell$ in ∇V_ℓ to the unique map $\vec{v} \to \vec{v}[v'_\ell / v_\ell]$ in $\mathbb{B}_{\vec{V}}$.

This captures exactly the "view update" paradigm in database theory: on the one hand, projecting from the state $\vec{v}$ to its component v_ℓ provides a *view* on the data encoded by $\vec{v}$; while lifting the morphism $v_\ell \to v'_\ell$ to one $\vec{v} \to \vec{v}[v'_\ell / v_\ell]$ encodes *updating* the state in light of the given update of the view. The pleasant feature here is that all of this is completely automatic once we specify the way in which V_ℓ-valued state is to be interpreted into $\vec{V}$-valued store.

8.5. Tape

Our final example is a variant on a particular case of the previous one; it was introduced *qua* monad in [14], with the presentation given here due to [15] and, independently, [27]. Given a set V, we consider the constant $\mathbb{Z}$-indexed family of sets $V^{(\mathbb{Z})} = (V : \ell \in \mathbb{Z})$. The theory of a *$V$-valued tape* is obtained by augmenting the theory of $V^{(\mathbb{Z})}$-valued store with two new unary operations right and right^{-1} satisfying the axioms $\mathsf{right}^{-1}(\mathsf{right}(x))$, $\mathsf{right}(\mathsf{right}^{-1}(x)) \equiv x$, and $\mathsf{right}(\mathsf{put}_u^{(\ell)}(x)) \equiv \mathsf{put}_u^{(\ell+1)}(\mathsf{right}(x))$ for all $\ell \in \mathbb{Z}$. By arguing much as before, we see that this last axiom implies also that $\mathsf{right}(\mathsf{get}^{(\ell)}(x)) \equiv \mathsf{get}^{(\ell+1)}(\lambda v.\, \mathsf{right}(x_v))$.

This theory encapsulates interaction with a doubly-infinite tape, each of whose locations $\ell \in \mathbb{Z}$ contains a value in V which can be read or updated via get and put, and whose head position may be moved right or left via right and right^{-1}. A comodel structure on a set S comprises a $\mathbb{Z}$-indexed family of commuting lens structures $(g^{(\ell)} : S \to V, p^{(\ell)} : V \times S \to S)$ together with a bijection $r : S \to S$ such that $r(p^{(\ell+1)}(u, \vec{v})) = p^{(\ell)}(u, r(\vec{v}))$ for each $\ell \in \mathbb{Z}$. It is easy to see that the final comodel of this theory is the final comodel $\boldsymbol{V}^{\mathbb{Z}}$ of $\vec{V}$-valued store, augmented with the co-operations $[\![\mathsf{right}]\!](\vec{v})_k = v_{k+1}$ and $[\![\mathsf{right}^{-1}]\!](\vec{v})_k = v_{k-1}$. By similar calculations to before, we now find that

Proposition 8.6 *The behaviour category of the theory of V-valued tape has object-set $V^{\mathbb{Z}}$, while a map $\vec{v} \to \vec{w}$ is an integer i such that $\vec{v}_{(-)+i}$ and $\vec{w}$ differ in only finitely many places. The comodel classifying the behaviour $\vec{v} \in V^{\mathbb{Z}}$ has underlying set*

$$\{(i \in \mathbb{Z}, \vec{w} \in V^{\mathbb{Z}}) : \vec{v} \text{ and } \vec{w} \text{ differ in only finitely many positions}\}$$

with operations $[\![\mathsf{right}^{\pm 1}]\!](i, \vec{w}) = (i \pm 1, \vec{w})$, $[\![\mathsf{get}^{(\ell)}]\!](i, \vec{w}) = (w_{i+\ell}, (i, \vec{w}))$, *and* $[\![\mathsf{put}_v^{(\ell)}]\!](i, \vec{w}) = (i, \vec{w}[v / w_{i+\ell}])$.

If the set V comes endowed with a bijective pairing function $\langle -, - \rangle : V \times V \to V$, say with inverse $(p, q) : V \to V \times V$, then there is an interpretation f of the theory of V-valued reversible

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

input into the theory of V-valued tape, given by

$$\mathsf{read}^f(\lambda v.\, x_v) := \mathsf{get}^{(0)}(\lambda w.\, \mathsf{put}^{(0)}_{p(w)}(\mathsf{right}(x_{q(w)})))$$
$$(\mathsf{unread}_u)^f(x) := \mathsf{left}(\mathsf{get}^{(0)}(\lambda w.\, \mathsf{put}^{(0)}_{\langle w,u\rangle}(x)))\,.$$

This induces a cofunctor on behaviour categories that acts as follows.

- On objects, for each $\vec{v} \in V^{\mathbb{Z}}$ in the behaviour category for a V-valued tape, we produce the object $f^*(\vec{v}) \in V^{\mathbb{N}}$ given by $f^*(\vec{v})_n = q(v_n)$;
- On morphisms, if we are given $\vec{v} \in V^{\mathbb{Z}}$ and a map $i\colon f^*(\vec{v}) \to W$ in the behaviour category for V-valued reversible input then our cofunctor lifts this to the morphism $i\colon \vec{v} \to \vec{w}$ in the behaviour category of V-valued tape where

$$w_k = \begin{cases} v_{k+i} & \text{if } k < -i; \\ p(v_{k+i}) & \text{if } -i \leqslant k < 0; \\ \langle p(v_{k+i}), W_k\rangle & \text{if } 0 \leqslant k. \end{cases}$$

  Note that, since $i\colon f^*(\vec{v}) \to W$ in the behaviour category for V-valued reversible input (cf. Proposition 8.1), we know that for some $N \in \mathbb{N}$, we have $W_k = f^*(\vec{v})_{k+i} = q(v_{k+i})$ for all $k > N$. It follows that $w_k = \langle p(v_{k+i}), q(v_{k+i})\rangle = v_{k+i}$ for all $k > N$, so that $\vec{v}_{(-)+i}$ and $\vec{w}$ do indeed only differ in finitely many places, as required for i to be a well defined map $\vec{v} \to \vec{w}$ in the behaviour category of V-valued tape.

9. Examples and Applications: Costructure

In this final section, we illustrate our understanding of the costructure functor by providing some sample calculations of the behaviour categories associated with comonads on $\mathcal{S}\mathrm{et}$.

9.1. Coalgebras for polynomial endofunctors

For any endofunctor $F\colon \mathcal{S}\mathrm{et} \to \mathcal{S}\mathrm{et}$, we can consider the category F-coalg of *F-coalgebras*, i.e., sets X endowed with a map $\xi : X \to FX$. As is well known, for suitable choices of F, such coalgebras can model diverse kinds of automaton and transition system; see [34] for an overview.

If F is accessible, then the forgetful functor F-coalg $\to \mathcal{S}\mathrm{et}$ will have a right adjoint and be strictly comonadic, meaning that we can identify F-coalg with the category of Eilenberg–Moore coalgebras of the induced comonad Q_F on $\mathcal{S}\mathrm{et}$; in light of this, we call Q_F the *cofree comonad* on the endofunctor F. Explicitly, the values of the cofree comonad can be described via the greatest fixpoint formula

$$Q_F(V) = \nu X.\, V \times F(X)\,. \tag{50}$$

The objective of this section is to calculate the behaviour categories of cofree comonads Q_F for some natural choices of F. To begin with, let us assume that F is *polynomial* in the sense of Section 3.1, meaning that it is can be written as a coproduct of representables

$$F(X) = \textstyle\sum_{\sigma \in \Sigma} X^{|\sigma|} \tag{51}$$

for some set Σ and family of sets $(|\sigma| : \sigma \in \Sigma)$. In this case, as is well known, the fixpoint (50) can be described as a set of *trees*.

Definition 9.1 (F-trees). *Let F be a polynomial endofunctor (51).*

- *An* F-path of length k *is a sequence $P = \sigma_0 e_1 \sigma_1 \cdots e_k \sigma_k$ where each $\sigma_i \in \Sigma$ and each $e_i \in |\sigma_{i-1}|$.*

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

- *An* F-tree *is a subset T of the set of F-paths such that:*

  (i) T contains a unique path of length 0, written $* \in T$;
  (ii) If T contains $\sigma_0 e_1 \cdots e_k \sigma_k e_{k+1} \sigma_{k+1}$, then it contains $\sigma_0 e_1 \cdots e_k \sigma_k$;
  (iii) If T contains $\sigma_0 e_1 \cdots e_k \sigma_k$, then for each $e_{k+1} \in |\sigma_k|$, it contains a unique path of the form $\sigma_0 e_1 \cdots e_k \sigma_k e_{k+1} \sigma_{k+1}$.

- *If V is a set, then a V-labelling for an F-tree T is a function $\ell \colon T \to V$.*
- *If T is an F-tree and $P = \sigma_0 e_1 \cdots e_k \sigma_k \in T$, then T_P is the F-tree*

$$T_P = \{\sigma_k e_{k+1} \cdots e_m \sigma_m : \sigma_0 e_1 \cdots e_m \sigma_m \in T\} .$$

  If $\ell \colon T \to V$ is a labelling for T, then $\ell_P \colon T_P \to V$ is the labelling with $\ell_P(\sigma_k e_{k+1} \cdots e_m \sigma_m) = \ell(\sigma_0 e_1 \cdots e_m \sigma_m)$.

Lemma 9.2 *The cofree comonad Q_F on a polynomial F as in (51) is given as follows:*

- *$Q_F(V)$ is the set of V-labelled F-trees;*
- *The counit $\varepsilon_V \colon Q_F(V) \to V$ sends (T, ℓ) to $\ell(*) \in V$;*
- *The comultiplication $\delta_V \colon Q_F(V) \to Q_F Q_F(V)$ sends (T, ℓ) to $(T, \ell^\sharp)$, where $\ell^\sharp \colon T \to Q_F(V)$ sends P to (T_P, ℓ_P).* □

We may use this result to calculate the behaviour category $\mathbb{B}_F$ of the comonad Q_F. Clearly, objects of $\mathbb{B}_F$ are elements of $Q_F(1)$, i.e., (unlabelled) F-trees. Morphisms of $\mathbb{B}_F$ with domain T are, by definition, natural transformations $(Q_F)_T \Rightarrow \mathrm{id}$; but the functor $(Q_F)_T$ is visibly isomorphic to the representable functor $(-)^T$, so that by the Yoneda lemma, morphisms in $\mathbb{B}_F$ with domain T correspond bijectively with elements $P \in T$. Given this, we may easily read off the remainder of the structure in Definition 6.3 to obtain:

Proposition 9.3 *Let F be a polynomial endofunctor of* Set. *The behaviour category $\mathbb{B}_F$ of the cofree comonad Q_F has:*

- *Objects given by F-trees T;*
- *Morphisms $P \colon T \to T'$ are elements $P \in T$ such that $T_P = T'$;*
- *Identities are given by $1_T = * \colon T \to T$;*
- *Composition is given by $(\sigma_k e_{k+1} \cdots e_m \sigma_m) \circ (\sigma_0 e_1 \cdots e_k \sigma_k) = \sigma_0 e_1 \cdots e_m \sigma_m$.*

It is not hard to see that $\mathbb{B}_F$ is, in fact, the free category on a graph: the generating morphisms are those of the form $\sigma_0 e_1 \sigma_1$.

Remark 9.4 *When F is polynomial, the cofree comonad Q_F is again polynomial: indeed, we have $Q_F(V) \cong \sum_{T \in F\text{-tree}} V^T$. Thus, Q_F is a presheaf comonad, and it will follow from Proposition 7.9 below that it is in fact the presheaf comonad of the behaviour category $\mathbb{B}_F$. Thus, we arrive at the (not entirely obvious) conclusion that, for F polynomial, the category of F-coalgebras is equivalent to the presheaf category $[\mathbb{B}_F, \mathrm{Set}]$.*

Example 9.5 *Let E be an alphabet. A* deterministic automaton over E *is a set S of states together with a function $(t, h) \colon S \to S^E \times \{\top, \bot\}$. For a state $s \in S$, the value $h(s)$ indicates whether or not h is an accepting state, while $t(s)(e) \in S$ gives the state reached from s by transition along $e \in E$.*

Deterministic automata are F-coalgebras for the polynomial functor $F(X) = \sum_{a \in \{\bot, \top\}} X^E$. It is easy to see that, in this case, the set of F-trees can be identified with the power set $\mathcal{P}(E^*)$ via the assignment

$$T \mapsto \{e_1 \cdots e_n \in E^* : \sigma_0 e_1 \ldots \sigma_{n-1} e_n \top \in T\} .$$

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

In these terms, the behaviour category $\mathbb{B}_F$ can be identified with the free category on the graph whose vertices are subsets of E^*, and whose edges are $e\colon L \to \partial_e L$ for each $L \subseteq E^*$ $e \in E$, where $\partial_e L = \{e_1 \cdots e_n \in E^* : ee_1 \cdots e_n \in L\}$. Note that this is precisely the transition graph of the final deterministic automaton over E.

9.2. Coalgebras for non-polynomial endofunctors

When we consider cofree comonads over *non*-polynomial endofunctors F, things become more delicate. To illustrate this, let us consider the case of the *finite multiset* endofunctor

$$M(X) = \textstyle\sum_{n\in\mathbb{N}} X^n/\mathcal{S}_n\,.$$

An F-coalgebra is a non-deterministic weighted transition system with transition weights in the additive monoid of natural numbers. As in the preceding section, we have a description of the associated cofree comonad in terms of trees:

Definition 9.6 (Symmetric trees) *A* symmetric tree T *is a diagram of finite sets and functions*

$$\cdots \xrightarrow{\partial} T_n \xrightarrow{\partial} \cdots \xrightarrow{\partial} T_1 \xrightarrow{\partial} T_0$$

where $T_0 = \{*\}$. *We may write* $|T|$ *for the set* $\sum_k T_k$. *A* V*-labelling for a symmetric tree* T *is a function* $\ell\colon |T| \to V$. *Given a* V*-labelled tree* (T, ℓ) *and* $t \in T_k$, *we write* (T_t, ℓ_t) *for the labelled tree with* $(T_t)_n = \{u \in T_{n+k} : \partial^k(u) = t\}$, *and with* ∂*'s and labelling inherited from* T. *An isomorphism* $\theta\colon (T, \ell) \to (T', \ell')$ *of* V*-labelled trees is a family of invertible functions* $\theta_n\colon T_n \to T'_n$ *commuting with the* ∂*'s and the functions to* V. □

Lemma 9.7 *The cofree comonad on the finite multiset endofunctor* M *has:*

- $Q_M(V)$ *given by the set of isomorphism-classes of* V*-labelled symmetric trees;*
- *The counit* $\varepsilon_V\colon Q_M(V) \to V$ *given by* $(T, \ell) \mapsto \ell(*)$;
- *The comultiplication* $\delta_V\colon Q_M(V) \to Q_M Q_M(V)$ *given by* $(T, \ell) \mapsto (T, \ell^\sharp)$, *where* $\ell^\sharp\colon |T| \to Q_M(V)$ *sends* $t \in T_k$ *to* (T_t, ℓ_t).

Note that an alternative description of this cofree comonad may be extracted from [2, Theorem 6.11]; this describes the final M-coalgebra $Q_M(1)$, but the description may be easily adapted to one for an arbitrary $Q_M(V)$.

Given a symmetric tree T, we call $t \in T_k$ *rigid* if any automorphism of T fixes t.

Proposition 9.8 *The behaviour category of the cofree comonad* Q_M *has:*

- *Objects given by isomorphism-class representatives of symmetric trees* T;
- *Morphisms* $t\colon T \to T'$ *are rigid elements* $t \in T$ *such that* $T_t \cong T'$;
- *The identity on* T *is* $*\colon T \to T$;
- *The composite of* $t\colon T \to T'$ *and* $u\colon T' \to T''$ *is* $\theta(u)\colon T \to T''$, *where* θ *is any tree isomorphism* $T' \to T_t$.

Proof. Let us write $\mathsf{Q} = \mathsf{Q}_M$. Clearly the object-set $Q(1)$ of the behaviour category can be identified with a set of isomorphism-class representatives of symmetric trees. Now, for any such representative T, the subfunctor $Q_T \subseteq Q$ sends a set V to the set of all isomorphism-classes of V-labellings of T, which is easily seen to be the quotient $V^{|T|}/\mathrm{Aut}(T)$ of $V^{|T|}$ by the evident action of the group of tree automorphisms of T. Thus, by the Yoneda lemma, the set of natural transformations $Q_T \Rightarrow \mathrm{id}$ can be identified with the set of elements $t \in |T|$ which are fixed by the $\mathrm{Aut}(T)$ action, i.e., the rigid elements of T. For a given rigid element $t \in |T|$, the corresponding $Q_T \Rightarrow \mathrm{id}$

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

sends a V-labelling $\ell\colon |T| \to V$ in $Q_T(V)$ to $\ell(t) \in V$; and it follows that the unique factorisation in (32) is of the form $Q_T \Rightarrow Q_{T'}$ where $T' \cong T_t$. Tracing through the remaining aspects of the definition of behaviour category yields the result. □

For a similar example in this vein, we may calculate the behaviour category of the cofree comonad generated by the finite powerset functor P_f. In this case, things are even more degenerate: the behaviour category turns out to be the *discrete* category on the final P_f-coalgebra.

9.3. Local homeomorphisms

For our final example, we compute the behaviour category of the comonad classifying local homeomorphisms over a topological space.

Definition 9.9 (Reduced power) *If A, X are sets and $\mathcal{F}$ is a filter of subsets of X, then two maps $\varphi, \psi\colon X \to A$ are* $\mathcal{F}$-equivalent *when $\{x \in X : \varphi(x) = \psi(x)\} \in \mathcal{F}$. The* reduced power $A^{\mathcal{F}}$ *is the quotient of A^X by $\mathcal{F}$-equivalence.*

Definition 9.10 (Sheaf comonad) *Let X be a topological space. The* sheaf comonad Q_X *is the accessible comonad on* $\mathcal{S}\mathrm{et}$ *induced by the adjunction*

$$\mathcal{L}\mathrm{h}/X \underset{U}{\overset{C}{\underset{\longrightarrow}{\overset{\longleftarrow}{\top}}}} \mathcal{S}\mathrm{et}/X \underset{\Sigma}{\overset{\Delta}{\underset{\longrightarrow}{\overset{\longleftarrow}{\top}}}} \mathcal{S}\mathrm{et}\,, \tag{52}$$

where $\mathcal{L}\mathrm{h}/X$ is the category of local homeomorphisms over X, where U is the evident forgetful functor, and where C sends $p\colon A \to X$ to the space of germs of partial sections of p. If we write $\mathcal{N}_x$ for the filter of open neighbourhoods of $x \in X$, then then this comonad has $Q_X(A) = \sum_{x \in X} A^{\mathcal{N}_x}$, and counit and comultiplication

$$\begin{array}{ll} \varepsilon_A\colon \sum_x A^{\mathcal{N}_x} \to A & \delta_A\colon \sum_x A^{\mathcal{N}_x} \to \sum_x \left(\sum_{x'} A^{\mathcal{N}_{x'}}\right)^{\mathcal{N}_x} \\ \quad (x, \varphi) \mapsto \varphi(x) & \quad (x, \varphi) \mapsto \left(x, \lambda y.\, (y, \varphi)\right). \end{array} \tag{53}$$

The adjunction in (52) is in fact strictly comonadic, so that we can identify the category of Q_X-coalgebras with the category of local homeomorphisms (= sheaves) over X.

Proposition 9.11 *Let X be a topological space. The behaviour category of the sheaf comonad Q_X is the poset $(X, \leqslant)$ of points of X under the specialisation order: thus $x \leqslant y$ just when every open set containing x also contains y.*

Proof. Writing Q for Q_X, we clearly have $Q(1) = X$, so that objects of the behaviour category are points of X. To characterise the morphisms with domain $x \in X$, we observe that the subfunctor $Q_x \subseteq Q$ is the reduced power functor $(-)^{\mathcal{N}_x}$, which can also be written as the directed colimit of representable functors $\mathrm{colim}_{U \in \mathcal{N}_x} (-)^U$. Thus, by the Yoneda lemma, the set of natural transformations $Q_x \Rightarrow \mathrm{id}$ can be identified with the filtered intersection $\bigcap_{U \in \mathcal{N}_x} U$, i.e., with the upset $\{y \in X : x \leqslant y\}$ of x for the specialisation order. Given $y \geqslant x$, the corresponding natural transformation $Q_x \Rightarrow \mathrm{id}$ has components $\varphi \mapsto \varphi(y)$; whence the unique factorisation in (32) is of the form $Q_x \Rightarrow Q_y$. By definition of behaviour category, we conclude that $\mathbb{B}_{\mathsf{Q}}$ is the specialisation poset of X. □

What we learn from this is that a local homeomorphism $p\colon S \to X$ can be seen as a coalgebraic structure with set of "states" S, with the "behaviour" associated with a state s given by $p(s)$, and with the possibility of transitioning uniformly from a state s of behaviour x to a state s' of behaviour y whenever $x \leqslant y$. This is intuitively easy to see: given $s \in S$ of behaviour x, we pick an open neighborhood $U \subseteq S$ mapping homeomorphically onto an open $V = p(U) \subseteq X$. Since $x = p(s) \in V$ and $x \leqslant y$, also $y \in V$ and so we may define s' of behaviour y to be $\left(\, p\big|_U \,\right)^{-1}(y) \in U$.

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

References

[1] Abbott, M., Altenkirch, T. and Ghani, N. (2005) Containers: Constructing strictly positive types. *Theoretical Computer Science* **342** 3–27.

[2] Adámek, J., Levy, P. B., Milius, S., Moss, L. S. and Sousa, L. (2015) On final coalgebras of power-set functors and saturated trees. *Applied Categorical Structures* **23** (4) 609–641.

[3] Aguiar, M. (1997) *Internal Categories and Quantum Groups*. PhD thesis, Cornell University.

[4] Ahman, D. and Bauer, A. (2020) Runners in action. In: *Programming Languages and Systems*, vol. 12075, Lecture Notes in Computer Science, Springer, 29–55.

[5] Ahman, D., Chapman, J. and Uustalu, T. (2012) When is a container a comonad? In: *Foundations of Software Science and Computational Structures*, vol. 7213, Lecture Notes in Computer Science, Heidelberg: Springer, 74–88.

[6] Ahman, D. and Uustalu, T. (2014) Coalgebraic update lenses. In : *Proceedings of the 30th Conference on the Mathematical Foundations of Programming Semantics (MFPS XXX)*, vol. 308, Electronic Notes in Theoretical Computer Science, Amsterdam: Elsevier Sci. B. V., 25–48.

[7] Ahman, D. and Uustalu, T. (2014) Update monads: Cointerpreting directed containers. In: *19th International Conference on Types for Proofs and Programs*, vol. 26 LIPIcs, Leibniz International Proceedings in Informatics, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 1–23.

[8] Ahman, D. and Uustalu, T. (2017) Taking updates seriously. In: *Proceedings of the 6th International Workshop on Bidirectional Transformations* (2017), CEUR Workshop Proceedings, 59–73.

[9] Barr, M. and Wells, C. (1985) *Toposes, Triples and Theories*, vol. 278, Grundlehren der Mathematischen Wissenschaften. Springer.

[10] Diers, Y. (1978) Spectres et localisations relatifs à un foncteur. *Comptes rendus hebdomadaires des séances de l'Académie des sciences* **287** 985–988.

[11] Dubuc, E. J. (1970) *Kan Extensions in Enriched Category Theory*, vol. 145, Lecture Notes in Mathematics. Springer.

[12] Foster, J. N., Greenwald, M. B., Moore, J. T., Pierce, B. C. and Schmitt, A. (2007) Combinators for bidirectional tree transformations: A linguistic approach to the view-update problem. *ACM Transactions on Programming Languages and Systems* 29 17–es.

[13] Goncharov, S. (2013) Trace semantics via generic observations. In: *Algebra and Coalgebra in Computer Science*, vol. 8089, Lecture Notes in Computer Science, Heidelberg: Springer, 158–174.

[14] Goncharov, S., Milius, S. and Silva, A. (2014) Towards a coalgebraic chomsky hierarchy. In: *TCS 2014, Rome* (2014), vol. 8705, Lecture Notes in Computer Science, Springer, 265–280.

[15] Goncharov, S., Milius, S. and Silva, A. (2020) Toward a uniform theory of effectful state machines. *ACM Transactions on Computational Logic* **21**.

[16] Higgins, P. J. and Mackenzie, K. C. H. (1993) Duality for base-changing morphisms of vector bundles, modules, Lie algebroids and Poisson structures. *Mathematical Proceedings of the Cambridge Philosophical Society* **114** 471–488.

[17] Johnstone, P. T. (1990) Collapsed toposes and Cartesian closed varieties. *Journal of Algebra* **129** 446–480.

[18] Jónsson, B. and Tarski, A. (1961) On two properties of free algebras. *Mathematica Scandinavica* **9** 95–101.

[19] Katsumata, S., Rivas, E. and Uustalu, T. (2019) Interaction laws of monads and comonads. Preprint, available as arXiv:1912.13477.

[20] Kelly, G. M. and Power, A. J. (1993) Adjunctions whose counits are coequalizers, and presentations of finitary enriched monads. *Journal of Pure and Applied Algebra* **89** 163–179.

[21] Kupke, C. and Leal, R. A. (2009) Characterising behavioural equivalence: Three sides of one coin. In: *Algebra and Coalgebra in Computer Science*, vol. 5728, Lecture Notes in Computer Science. Springer, 97–112.

[22] Lawvere, F. W. (1963) *Functorial Semantics of Algebraic Theories*. PhD thesis, Columbia University.

[23] Manes, E. (1976) *Algebraic Theories*, vol. 26. Graduate Texts in Mathematics. Springer.

[24] Møgelberg, R. E. and Staton, S. (2014) Linear usage of state. *Logical Methods in Computer Science* **10** 1:17, 52.

[25] Moggi, E. (1991) Notions of computation and monads. *Information and Computation* **93** 55–92.

[26] Pattinson, D. and Schröder, L. (2015) Sound and complete equational reasoning over comodels. In: *The 31st Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXI)*, vol. 319, Electronic Notes in Theoretical Computer Science. Elsevier, 315–331.

[27] Pattinson, D. and Schröder, L. (2016) Program equivalence is coinductive. In: *Proceedings of the 31st Annual ACM-IEEE Symposium on Logic in Computer Science (LICS 2016)* (2016), ACM, 10.

[28] Plotkin, G. and Power, J. (2001) Adequacy for algebraic effects. In: *Foundations of Software Science and Computation Structures (Genova, 2001)*, vol. 2030, Lecture Notes in Computer Science, Berlin: Springer, 1–24.

[29] Plotkin, G. and Power, J. (2002) Notions of computation determine monads. In: *Foundations of Software Science and Computation Structures (Grenoble, 2002)*, vol. 2303, Lecture Notes in *Computer Science*. Springer, Berlin, **2002**, pp. 342–356.

[30] Plotkin, G. and Power, J. (2003) Algebraic operations and generic effects. *Applied Categorical Structures* **11** 69–94.

[31] Plotkin, G. and Power, J. (2008) Tensors of comodels and models for operational semantics. *Electronic Notes in Theoretical Computer Science* **218** 295–311.

[32] Power, J. and Shkaravska, O. (2004) From comodels to coalgebras: State and arrays. In: *Proceedings of the Workshop on Coalgebraic Methods in Computer Science* (2004), vol. 106. Electronic Notes in Theoretical Computer Science, Elsevier, 297–314.

https://doi.org/10.1017/S0960129521000219 Published online by Cambridge University Press

[33] Renault, J. (1980) *A Groupoid Approach to C*-Algebras*. vol. 793. Lecture Notes in Mathematics, Berlin: Springer.
[34] Rutten, J. J. M. M. (2000) Universal coalgebra: A theory of systems. *Theoretical Computer Science* **249** 3–80.
[35] Rutten, J. J. M. M. and Turi, D. (1993) On the foundations of final semantics: nonstandard sets, metric spaces, partial orders. In: *Semantics: Foundations and Applications (Beekbergen, 1992)*, vol. 666, Lecture Notes in Computer Science, Berlin: Springer, 477–530.
[36] Thielecke, H. (1997) *Categorical Structure of Continuation Passing Style*. PhD thesis, University of Edinburgh.
[37] Uustalu, T. (2015) Stateful runners of effectful computations. In: *The 31st Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXI)*, vol. 319, Electronic Notes in Theoretical Computer Science, Amsterdam: Elsevier Sci. B. V., 403–421.

Cite this article: Garner R (2022). The costructure–cosemantics adjunction for comodels for computational effects. *Mathematical Structures in Computer Science* **32**, 374–419. https://doi.org/10.1017/S0960129521000219