

Taut Monads, Dynamic Logic and Determinism

Ernie Manes ¹

*Department of Mathematics and Statistics
University of Massachusetts
Amherst, MA 01003, USA*

Abstract

Studied are Kleisli categories of monads of sets which satisfy two properties motivated by functional properties of collections. Such categories have box and diamond operators which follow the laws of (loop-free) dynamic logic. A theorem of Kozen states that the category of sets and relations is complete for loop-free dynamic logic. It is shown that the Kleisli category of the filter monad is likewise complete. A morphism α is deterministic if $\langle \alpha \rangle Q \subset [\alpha]Q$ for all Q . Each “output value” αx is associated with a filter which is forced to be an ultrafilter when α is deterministic and αx is defined. Early work in the theory of domains abstracted from the partially ordered set of partial functions between two sets, ordered by extension. A different abstraction, suited to conditional constructs rather than recursive fixed point equations, is the notion of a locally Boolean poset, and this is used to compare restriction categories with deterministic Kleisli categories. The laws of dynamic logic in terms of $[\alpha]Q$ and $\langle \alpha \rangle Q$ for a single α hold in any topological space with $[\alpha]Q$ the interior operator and $\langle \alpha \rangle Q$ the closure operator.

Keywords: dynamic logic, collection type, taut monad, Boolean category, deterministic map, locally Boolean poset, filter, restriction category

1 Introduction

In this paper, a monad \mathbf{T} in the category **Set** of sets and (total) functions induces three constructs of interest. The first is the Kleisli category $\mathbf{Set}_{\mathbf{T}}$ which may be thought of as the category in which expressions are evaluated when the monad class \mathbf{T} has been activated. The second is the category $\mathbf{Set}^{\mathbf{T}}$ of algebras over the monad, which is used to determine if the monad is of collection type. The third is a class of topological spaces determined by certain endomorphisms of the Kleisli category whose box and diamond operators are the interior and closure operators.

A function $f : X \rightarrow Y$ maps a “point” x to another point fx . When the outputs are more general “distributions” on Y , write the set of such distributions as TY and consider functions of form $f : X \rightarrow TY$. Carrying this line of thought to

¹ Email: manes@math.umass.edu

its logical conclusion leads to the Kleisli category of a monad in **Set**. [10] is the first paper to classify monads from this point of view. More recently, programming interpretations have been given to such monads, “notions of computation” in [16] and monad classes (as in Haskell) in [21] where one “keeps the code and changes the monad on the fly”.

In three papers [12,13,15], the author has explored monads suitable to describe collection data types. A collection is a structure which holds values. A *deterministic* collection is such a structure which holds at most one value, and this paper studies such *deterministic monads*. In the finitary case, there are few surprises. In the infinitary case, however, the most generalized “deterministic value” (as is forced by the theory, not by definition) is an ultrafilter. The interpretation of an ultrafilter as a “generalized point” dates to the first paper about them [19].

But let us begin at the beginning.

2 Monads as Collection Types

We assume familiarity with the definition of a monad $\mathbf{T} = (T, \eta, \mu)$ with $id \xrightarrow{\eta} T \xleftarrow{\mu} TT$ natural transformations subject to three equations and with the equivalent formulation $\mathbf{T} = (T, \eta, (\cdot)^\#)$ where $\alpha : X \rightarrow TY \mapsto \alpha^\# : TX \rightarrow TY$ subject to three equations. (Monad basics can be found in [14], and uncited definitions and facts throughout the paper can be found there). The Kleisli category $\mathbf{Set}_\mathbf{T}$ has sets as objects, functions $X \rightarrow TY$ as morphisms $X \rightarrow Y$ with composition $\beta \circ \alpha = X \xrightarrow{\alpha} TY \xrightarrow{\beta^\#} TZ$ and η_X as the identity morphism of X . That this forms a category is the essence of the axioms on a monad. Recall that $\alpha^\# = TX \xrightarrow{T\alpha} TTY \xrightarrow{\mu_Y} TY$ and that $\mu_X = (id_{TX})^\#$ and for $f : X \rightarrow Y$, $Tf = (f^\circ)^\#$ where $f^\circ = X \xrightarrow{f} Y \xrightarrow{\eta_Y} TY$. For Lemma 2.24 and Proposition 5.7, we recall that a monad map $\lambda : (S, \eta, (\cdot)^\#) \rightarrow (T, \eta, (\cdot)^\#)$ is characterized by the equations $\eta\lambda = \eta$ and the following commutative squares induced by $\alpha : X \rightarrow SY$:

$$\begin{array}{ccc} SX & \xrightarrow{\lambda_X} & TX \\ \alpha^\# \downarrow & & \downarrow (\lambda_Y \alpha)^\# \\ SY & \xrightarrow{\lambda_Y} & TY \end{array}$$

Example 2.1 The list monad is $\mathbf{L} = (L, \eta, \mu)$ with LX the set of all lists $[x_1, \dots, x_n]$, ($n \geq 0$), with $x_i \in X$; $(Lf)[x_1, \dots, x_n] = [fx_1, \dots, fx_n]$; $\eta_X(x) = [x]$; $\mu_X[w_1, \dots, w_n]$ is the concatenation $w_1 \# \dots \# w_n$. For $\alpha : X \rightarrow LY$, $\alpha^\#[x_1, \dots, x_n]$ is obtained by substituting $\alpha(x_i)$ for x_i and then flattening.

Aspects of the monad structure are subject to well-known functional programming interpretations and we discuss two such aspects using the list monad as a specific example. Lf is $Map f$ and we generally give this interpretation to Tf . Similarly, μ_X is thought of as a “flatten” operator. Not every monad would behave correctly with respect to such interpretations about collections. One imagines that there is a large wish list of potential axioms for a useful theory. Surprisingly,

we shall require only two such axioms. In what follows, we write $2 = \{0, 1\}$ with $0 = \text{False}$, $1 = \text{True}$. Thus the characteristic function $\chi_A : X \rightarrow 2$ of $A \subset X$ satisfies $\chi_A^{-1}(1) = A$. The two properties can loosely be expressed as follows.

- (1) Given $A \subset X$, there exists a 2-valued functional expression to determine if $\tau \in TX$ holds only elements of A .
- (2) Given that $\tau \in T2$ holds elements in $A \subset T2$ and that each element of A holds only the value True , then $\mu_2(\tau)$ holds only the value True .

That these properties hold for the list monad is easily seen. Consider Boolean “and” $\wedge : L2 \rightarrow 2$ with $\wedge[x_1, \dots, x_n] = 1 \Leftrightarrow x_i = 1$ for all i . Then $LX \xrightarrow{L(\chi_A)} L2 \xrightarrow{\wedge} 2$ is true of $[x_1, \dots, x_n]$ if and only if each $x_i \in A$. Thus the desired functional expression to establish (1) is that which maps χ_A on the input list and then folds the result with “and”. (2) asserts that each symbol in $w_1 \# \dots \# w_n$ is True if and only if this holds for each w_i .

It is not hard to formalize (1,2) as requirements for an arbitrary monad of **Set**. T preserves monics (even empty ones) [13, Page 84]. Hence for $A \subset X$, “ $TA \subset TX$ ” so that TA formalizes $\{\tau \in TX : \tau \text{ holds only elements of } A\}$. In particular, $\psi = \chi_{T1} : T2 \rightarrow 2$ formalizes the elements of $T2$ which hold only 1 (where, recall, $1 = \text{True}$). Thus the functional expression to determine if $\tau \in TX$ holds only elements in $A \subset X$ is $TX \xrightarrow{T(\chi_A)} T2 \xrightarrow{\psi} 2$ and the formal statement that this expression works is that the squares

$$\begin{array}{ccc} TA & \xrightarrow{\quad} & \star \\ Ti \downarrow & & \downarrow 1 \\ TX & \xrightarrow{\psi T(\chi_A)} & 2 \end{array} \quad (3)$$

(which always commute) are pullbacks. (Here, $A \subset X$ has inclusion $i : A \rightarrow X$. We use \star as a one-element set rather than 1 to avoid confusion with True). For $A = \{1\} \subset 2$, $\chi_A = \text{id}$ so if (3) is always a pullback, the square

$$\begin{array}{ccc} T\star & \xrightarrow{\quad} & 1 \\ T(\text{True}) \downarrow & & \downarrow \text{True} \\ T2 & \xrightarrow{\psi} & 2 \end{array}$$

in particular is, that is in general,

$$(4) \quad \psi = \chi_{T1}$$

(2) may be expressed by saying that for $\tau \in TT2$, $\mu_2(\tau) \in T1 \Leftrightarrow (T\chi_{T1})\tau \in T1$, that is, the square

$$\begin{array}{ccc}
TT2 & \xrightarrow{T\psi} & T2 \\
\mu_2 \downarrow & & \downarrow \psi \\
T2 & \xrightarrow{\psi} & 2
\end{array} \quad (5)$$

commutes.

We are now ready to talk about the category $\mathbf{Set}^{\mathbf{T}}$ of algebras of the monad \mathbf{T} . See [9,14] for basic facts. The objects are all (X, ξ) with $\xi : TX \rightarrow X$ the **structure map** satisfying the two equations at the left, with maps called **T-homomorphisms** satisfying the square on the right:

$$\begin{array}{ccccc}
X & \xrightarrow{\eta_X} & TX & & TTX & \xrightarrow{T\xi} & TX & & TX & \xrightarrow{Tf} & TY \\
& \searrow id & \downarrow \xi & & \downarrow \mu_X & & \downarrow \xi & & \downarrow \xi & & \downarrow \theta \\
& & X & & TX & \xrightarrow{\xi} & X & & X & \xrightarrow{f} & Y
\end{array} \quad (6)$$

By the monad laws for \mathbf{T} , (TX, μ_X) is a \mathbf{T} -algebra. It is freely generated by X (in the same sense as one talks about free monoids, groups and modules), since for each function $f : X \rightarrow Y$ with (Y, θ) a \mathbf{T} -algebra, $f^\# = TX \xrightarrow{Tf} TY \xrightarrow{\theta} Y$ is the unique \mathbf{T} -homomorphism with $f^\# \eta_X = f$. The earlier $\alpha^\# : TX \rightarrow TY$ is a special case of this construction. Thus to prove that two \mathbf{T} -homomorphisms $f, g : (TX, \mu_X) \rightarrow (Y, \theta)$ are equal it suffices to show equality restricted to the “generators” η_X , that is, that $f\eta_X = g\eta_X$. By the naturality of μ , $Tf : (TX, \mu_X) \rightarrow (TY, \mu_Y)$ is a \mathbf{T} -homomorphism for any $f : X \rightarrow Y$. By the middle square in (6), every structure map is a \mathbf{T} -homomorphism. This proves directly that $f^\# : (TX, \mu_X) \rightarrow (Y, \theta)$ is a \mathbf{T} -homomorphism.

The categories of form $\mathbf{Set}^{\mathbf{T}}$ are precisely the equationally definable classes of universal algebras (with not necessarily finitary operations) with the only exceptions being situations such as complete lattices whose free algebras are “too large to have a cardinal number”.

Say that a monad \mathbf{T} is **trivial** if it has no algebra with two or more elements. In particular, $|TX| \leq 1$. There are only two trivial monads, easily discovered by the reader, and we will henceforth work only with non-trivial monads. In that case, by taking cartesian products in the category of algebras, there exist algebras of arbitrarily large cardinality.

Lemma 2.2 *For a non-trivial monad $\mathbf{T} = (T, \eta, \mu)$, if $a \neq b \in X$ then $\eta_X(a) \notin Tb$ in TX . (We abuse notation and write Ta for $T\{a\}$, here and elsewhere).*

Proof Let (Y, θ) be a \mathbf{T} -algebra with at least three elements and let $f, g : X \rightarrow Y$ be functions with fa, fb, ga distinct and $fb = gb$. Then $f^\#, g^\# : (TX, \mu_X) \rightarrow (Y, \theta)$ agree on Tb but disagree on $\eta_X(a)$ so $\eta_X(a) \notin Tb$. \square

Corollary 2.3 *For a non-trivial monad, η_X is componentwise monic.* \square

In [13, Definition 3.1] a **support classifier** for a monad \mathbf{T} was defined to be a \mathbf{T} -algebra $(2, \psi)$ satisfying that (3) is always a pullback. In that case, (4,5) hold since (4) follows from (3) being a pullback and (5) is one of the \mathbf{T} -algebra equations in (6). Conversely, to show that (3)-pullback and (5) implies support classifier (with ψ defined by (4)) we need only show the remaining algebra law that $2 \xrightarrow{\eta_2} T2 \xrightarrow{\psi} 2 = id$. By Lemma 2.2, $0 \notin T1$ so $\psi\eta_2(0) = 0$. As $1 \in T1$, $\psi\eta_2(1) = 1$. Thus the axioms (3)-pullback and (5), motivated by functional programming concerns about collections, are equivalent to the existence of a support classifier.

Example 2.4 Let TX be the set of all formal real linear combinations $\lambda_1 x_1 + \dots + \lambda_n x_n$ with λ_i real and $x_i \in X$, $\eta_X(x) = 1x$, $\alpha^\#(\lambda_1 x_1 + \dots + \lambda_n x_n) = \lambda_1 \alpha(x_1) + \dots + \lambda_n \alpha(x_n)$, simplified in the usual way. The algebras are real vector spaces. Since there is no two-element real vector space, this monad has no support classifier.

We have motivated that \mathbf{T} should have a support classifier and we now reap the benefits of known theorems. For that, a few definitions are needed. An **inverse image square** is a pullback of $X \xrightarrow{f} Y \xleftarrow{j} Q$ with j monic. A functor F is **taut** if it maps inverse image squares to pullbacks (which are then necessarily inverse image squares). A natural transformation is **taut** if the naturality squares induced by monics are pullbacks. A monad (T, η, μ) is **taut** if T, η, μ all are.

Theorem 2.5 [13, Theorem 3.3] *A monad has a support classifier if and only if it is non-trivial and taut.* \square

In the interests of brief terminology, we will hence favor the term “taut monad” for a monad with a support classifier.

Example 2.6 The filter monad $\mathbf{F} = (F, \eta, \mu)$ is a taut monad. Though well known, we review the definition here. FX is the set of all filters on X (i.e. the set of all filters on the Boolean algebra 2^X) including the improper filter 2^X , $(Ff)\mathcal{F} = \{B \subset Y : f^{-1}B \in \mathcal{F}\}$, $\eta_X(x)$ is the principal ultrafilter $prin(x) = \{A \subset X : x \in A\}$, $\mu_X(\mathcal{H}) = \{A \subset X : \Box A \in \mathcal{H}\}$ (where $\Box A = \{\mathcal{F} \in FX : A \in \mathcal{F}\}$), $\alpha^\# \mathcal{F} = \{B \subset Y : \{x \in X : B \in \alpha x\} \in \mathcal{F}\}$. By the theorems of [6,22], the category of \mathbf{F} -algebras is continuous lattices with morphisms that preserve directed suprema and arbitrary infima, the structure map $\xi : FL \rightarrow L$ of the continuous lattice L being the lim-inf operation $\xi(\mathcal{F}) = \bigvee_{F \in \mathcal{F}} \bigwedge F$. \mathbf{F} is taut with support classifier 2 with $0 < 1$.

The filter monad is intimately connected with general taut monads as the next result shows.

Theorem 2.7 [13, theorem 3.3] *A monad \mathbf{T} is taut if and only if there exists a taut monad map $\mathbf{T} \rightarrow \mathbf{F}$. In that case, the **support map***

$$TX \xrightarrow{\text{supp}_X} FX, \quad \tau \mapsto \{A \subset X : \tau \in TA\}$$

defines a canonical taut monad map $\mathbf{T} \rightarrow \mathbf{F}$.

Any monad map $\lambda : \mathbf{S} \rightarrow \mathbf{T}$ for monads in a category \mathcal{K} induces a forgetful functor $\mathcal{K}^{\mathbf{T}} \rightarrow \mathcal{K}^{\mathbf{S}}$ defined by $(X, TX \xrightarrow{\xi} X) \mapsto (X, SX \xrightarrow{\lambda_X} TX \xrightarrow{\xi} X)$. Applying

this in the context of the previous theorem, we see that if L is any continuous lattice and \mathbf{T} is any taut monad, then $\xi : TL \rightarrow L$, $\xi(\omega) = \bigvee_{\omega \in TA} \bigwedge A$ is a \mathbf{T} -algebra. It is routine to check that if $L = 2$ with $0 < 1$, then the resulting \mathbf{T} -algebra is the support classifier of \mathbf{T} .

We now veer off in what seems at first to be a totally unrelated direction, but which will provide us with yet another characterization of taut monads. We consider how to interpret loop-free dynamic logic [7,8,18] in a Kleisli category.

Definition 2.8 A category \mathcal{B} is a **Boolean category** [11] if the following four axioms hold.

(B.1) \mathcal{B} has finite coproducts. The initial object is denoted 0.

(B.2) For $X \xrightarrow{f} Y \xleftarrow{j} Q$, with j a coproduct injection, there exists a pullback

$$\begin{array}{ccc} [f]Q & \xrightarrow{g} & Q \\ i \downarrow & & \downarrow j \\ X & \xrightarrow{f} & Y \end{array}$$

with i a coproduct injection.

(B.3) A coproduct injection pulls back a coproduct to a coproduct.

(B.4) If $X \xrightarrow{1} X \xleftarrow{1} X$ is a coproduct, then X is an initial object.

Proposition 2.9 [11, Proposition 5.15, Theorem 5.11] *The following hold in any Boolean category.*

(1) *Every coproduct injection is an equalizer. The subobjects of X represented by a coproduct injection are called **summands** of X and the class of such is denoted $\text{Summ}(X)$.*

(2) *$\text{Summ}(X)$ is a Boolean algebra with least element $0 \rightarrow X$, greatest element $1 : X \rightarrow X$, intersection via pullback and Boolean complement via (unique) coproduct complement. For $Q, R \in \text{Summ}(X)$, $Q \rightarrow Q \cup R \leftarrow R$ is a coproduct if and only if $Q \cap R = 0$.* \square

As explained in [11], a Boolean category is a semantic model of dynamic logic with box operator $[f]Q$ and diamond operator $\langle f \rangle Q = ([f]Q')'$. A number of examples will be given shortly.

For any monad \mathbf{T} in a category \mathcal{K} , there is a canonical functor $\mathcal{K} \rightarrow \mathcal{K}_{\mathbf{T}}$ which is the identity on objects and which maps $f : X \rightarrow Y$ to $f^{\diamond} : X \rightarrow TY$ (as defined at the beginning of this section). For a non-trivial monad in **Set**, η is pointwise monic, so this functor may be regarded as the inclusion of a subcategory. In general, this functor has a right adjoint, the functor which maps $\alpha : X \rightarrow Y$ in $\mathcal{K}_{\mathbf{T}}$ to $\alpha : X \rightarrow TY$ in \mathcal{K} . In particular, if $P \xrightarrow{i} X \xleftarrow{j} Q$ is a coproduct in \mathcal{K} , $P \xrightarrow{i^{\diamond}} X \xleftarrow{j^{\diamond}} Q$ is a coproduct in $\mathcal{K}_{\mathbf{T}}$. We now come to a subtle point. Even though coproducts in a category are unique up to isomorphism, the sub-poset of summands in \mathcal{K} and $\mathcal{K}_{\mathbf{T}}$ can be quite different. First observe, that for every non-trivial monad, the Kleisli category is equivalent to the category of free algebras via

$\alpha : X \rightarrow TY \mapsto \alpha^\# : (TX, \mu_X) \rightarrow (TY, \mu_Y)$. Consider the real vector space monad of Example 2.4. The Kleisli category of this monad is equivalent to the category of algebras, all real vector spaces, because every vector space has a basis. The only nondegenerate coproduct decomposition of 2 is $\star \xrightarrow{True} 2 \xleftarrow{False} \star$. Now, $T2$ is the plane and the induced coproduct decomposition of the plane is as the direct sum of the x -axis and the y -axis. But any pair of distinct lines through the origin gives a coproduct decomposition in the Kleisli category. The four-element Boolean algebra 2^2 has been enlarged to a non-Boolean infinite poset of summands. From our current point of view, the summands are the tests of the assertion logic. In adding a monad “on the fly” we want the new assertion logic to apply to the same tests. But here we get ahead of ourselves, so let us turn to the requisite definition and theorem.

Definition 2.10 For a monad \mathbf{T} in \mathcal{K} , say that **summands are standard** if for every coproduct $P \xrightarrow{\alpha} X \xleftarrow{\beta} Q$ in $\mathcal{K}_{\mathbf{T}}$ there exists a coproduct $A \xrightarrow{i} X \xleftarrow{j} B$ in \mathcal{K} and an isomorphism φ in $\mathcal{K}_{\mathbf{T}}$ such that the triangle

$$\begin{array}{ccc} A & \xrightarrow{i^\diamond} & X \\ \varphi \searrow & & \nearrow \alpha \\ & P & \end{array}$$

commutes in $\mathcal{K}_{\mathbf{T}}$.

In the vector space example, summands are not standard. If in the triangle above, A is the x -axis and P is the line $y = x$, then A and P are isomorphic vector spaces but no isomorphism between them makes the triangle commute.

We are now ready to give our final characterization theorem of general taut monads.

Theorem 2.11 [15, Theorem 9.2] *A non-trivial monad \mathbf{T} is taut if and only if $\mathbf{Set}_{\mathbf{T}}$ is a Boolean category for which the functor $\mathbf{Set} \rightarrow \mathbf{Set}_{\mathbf{T}}$ is taut and for which summands are standard.* \square

Any endomorphism $\alpha : X \rightarrow TX$ in $\mathbf{Set}_{\mathbf{T}}$ is a “one-object model” of dynamic logic. This generalizes the Kripke models first introduced (see e.g. [8]).

The introduction of the box operator in $\mathbf{Set}_{\mathbf{T}}$ creates the need to better understand these pullbacks. This problem has a straightforward solution as follows.

Proposition 2.12 *Let \mathbf{T} be a taut monad, and let $\alpha : X \rightarrow TY$, $P \subset X$ with inclusion $i : P \rightarrow X$, $Q \subset Y$ with inclusion $j : Q \rightarrow Y$. Consider the squares*

$$\begin{array}{ccccc} P & \xrightarrow{i^\diamond} & X & & TP & \xrightarrow{Ti} & TX & & P & \xrightarrow{i} & X \\ \beta \downarrow & (A) & \downarrow \alpha & & \beta^\# \downarrow & (B) & \downarrow \alpha^\# & & \beta \downarrow & (C) & \downarrow \alpha \\ Q & \xrightarrow{j^\diamond} & Y & & TQ & \xrightarrow{Tj} & TY & & TQ & \xrightarrow{Tj} & TY \end{array}$$

where (A) is in $\mathbf{Set}_{\mathbf{T}}$ and (B, C) are in \mathbf{Set} . Then if any square commutes, they all do. If any square is a pullback, they all are.

Proof The result for (A,B) holds even more generally and for any monad in any category [11, Lemma 6.10 (2)]. Now consider the diagram

$$\begin{array}{ccccc} P & \xrightarrow{\eta_P} & TP & \xrightarrow{\beta^\#} & TQ \\ i \downarrow & (D) & \downarrow Ti & (B) & \downarrow Tj \\ X & \xrightarrow{\eta_X} & TX & \xrightarrow{\alpha^\#} & TY \end{array}$$

whose perimeter is (C). As the naturality square (D) commutes, (C) commutes if (B) does. Conversely, both paths in (B) are homomorphisms from the free algebra (TP, μ_P) so (B) commutes if it commutes restricted to the “generators” η_P , that is, (B) commutes if (C) does. Since (D) is a pullback as η is taut, (C) is a pullback if (B) is. Conversely, as $\alpha^\# = \mu_Y(T\alpha)$, $\beta^\# = \mu_Q(T\beta)$, the diagram

$$\begin{array}{ccccc} TP & \xrightarrow{T\beta} & TTP & \xrightarrow{\mu_Q} & TQ \\ Ti \downarrow & & \downarrow TTi & & \downarrow Tj \\ TX & \xrightarrow{T\alpha} & TTY & \xrightarrow{\mu_Y} & TY \end{array}$$

and the tautness of T and μ give that (B) is a pullback if (C) is. \square

We now use the tools from Theorems 2.7, 2.11 to obtain a useful result:

Proposition 2.13 For \mathbf{T} a taut monad, $\alpha : X \rightarrow TY$, $Q \subset Y$,

$$\begin{aligned} [\alpha]Q &= \{x \in X : Q \in \text{supp}_Y(\alpha x)\} \\ \langle \alpha \rangle Q &= \{x \in X : Q' \notin \text{supp}_Y(\alpha x)\} \end{aligned}$$

(where $Q' = Y \setminus Q$ is the complement of Q in Y).

Proof Because summands are standard in \mathbf{Set}_T , if (A) in Proposition 2.12 is a pullback, it constructs $P = [\alpha]Q$ in \mathbf{Set}_T . By that proposition, the pullback (C) in \mathbf{Set} has $P = [\alpha]Q$ so that $x \in [\alpha]Q \Leftrightarrow \alpha x \in TQ \Leftrightarrow Q \in \text{supp}_Y(\alpha x)$. Thus $x \in \langle \alpha \rangle Q \Leftrightarrow x \in ([\alpha]Q')' \Leftrightarrow x \notin [\alpha]Q' \Leftrightarrow Q' \notin \text{supp}_Y(\alpha x)$. \square

We round out the section by giving a number of examples and discussing their properties.

Example 2.14 Let E be a set of “exceptions”. Then (T, η, μ) is a taut monad if $TX = X + E$, $\eta_X = X \xrightarrow{\text{in}_X} X + E$, $\mu_X = X + E + E \xrightarrow{\langle \text{in}_X, \text{in}_E, \text{in}_E \rangle} X + E$. In the Kleisli category, a program behavior $\alpha : X \rightarrow Y + E$ either maps x to an output $y \in Y$ or raises an exception $e \in E$. (If divergence is considered as an exception in E , it is not “raised” of course). For $\beta : Y \rightarrow Z + E$, $\beta^\# = \langle \alpha, \text{in}_E \rangle$ so $(\beta \circ \alpha)x = e \in E$ if $\alpha x = e$; that is, an exception “sticks”. For $Q \subset Y$ we have

$$\begin{aligned} [\alpha]Q &= \{x \in X : \alpha x \in Q + E\} \\ \langle \alpha \rangle Q &= \{x \in X : \alpha x \in Q\} \end{aligned}$$

An algebra is a pair (X, p) where X is a set and $p : E \rightarrow X$ is any function. Homomorphisms $f : (X, p) \rightarrow (Y, q)$ satisfy $ft = q$. The support classifier is $(2, p)$ with $p(e) = \text{True}$ for all e . When E is empty, the Kleisli category is \mathbf{Set} . When E has one element, the Kleisli category is the category \mathbf{Pfn} of sets and partial

functions, and the category of algebras is the category of sets with base point. Here the box and diamond operators are

$$\begin{aligned} [\alpha]Q &= \{x \in X : \text{if } \alpha x \text{ is defined then } \alpha x \in Q\} \\ <\alpha>Q &= \{x \in X : \alpha x \text{ is defined and } \alpha x \in Q\} \end{aligned}$$

We call the case when E has one element the **exception monad** and write it $\mathbf{Ex} = (Ex, \eta, \mu)$.

Example 2.15 The **power set monad** is the taut monad given by $\mathbf{P} = (P, \eta, \mu)$ has $PX = 2^X$, $(Pf)A = fA = \{fa : a \in A\}$, $\eta_X(x) = \{x\}$, $\mu_X(\mathcal{A}) = \bigcup \mathcal{A}$, $\alpha^\# A = \bigcup_{a \in A} \alpha(a)$. The Kleisli category is isomorphic to the category **Rel** of sets and relations where $\alpha : X \rightarrow 2^Y$ corresponds to the relation $R \subset X \times Y$ via $x R y \Leftrightarrow y \in \alpha(x)$. For $R \subset X \times Y$, $S \subset Y \times Z$, the composition that corresponds to Kleisli composition is the usual one, $x S R z \Leftrightarrow \exists y, x R y, y S z$. The algebras of this monad are complete sup-semilattices where the structure map $\xi : PX \rightarrow X$ is the supremum. Thus $x \leq y \Leftrightarrow \xi\{x, y\} = y$. The support classifier is 2 with $0 < 1$. The box and diamond operators here are

$$\begin{aligned} [\alpha]Q &= \{x \in X : \alpha x \subset Q\} \\ <\alpha>Q &= \{x \in X : \alpha x \cap Q \neq \emptyset\} \end{aligned}$$

Rel is the standard model of multi-valued semantics. One can choose submonads to enforce further properties, e.g. non-empty subsets or finite subsets; these are submonads because a singleton is non-empty and finite and because a non-empty union of non-empty sets is non-empty and a finite union of finite sets is finite.

Example 2.16 We continue the discussion of the list monad of Example 2.1. The algebras are monoids, where the structure map $\xi : LX \rightarrow X$ of a monoid $(X, \cdot, 1)$ is $\xi[x_1, \dots, x_n] = x_1 \cdots x_n$. Conversely, given an algebra (X, ξ) , define $x \cdot y = \xi[x, y]$ with unit $\xi[\]$. This is a taut monad; the support classifier is 2 with unit *True* and multiplication \wedge , as was central to our earlier motivations. The box and diamond operators are defined by

$$\begin{aligned} [\alpha]Q &= \{x \in X : \alpha x \in LQ\} \quad (\text{possibly } \alpha x \text{ empty}) \\ <\alpha>Q &= \{x \in X : \alpha x \text{ has a symbol in } Q\} \end{aligned}$$

Example 2.17 Define the **probability monad** $\mathbf{Pr} = (Pr, \eta, \mu)$ as follows. $Pr X$ is the set of all countable-support probability distributions $\sum_x \lambda_x x$ with $\lambda_x \in [0, 1]$ the real unit interval, and $\sum \lambda_x = 1$. “Countable support” means that $\lambda_x \neq 0$ for at most countably many x . This is a submonad of the real-vectorspace monad of Example 2.4. To see this, consider $\alpha^\#(\sum_x \lambda_x x) = \sum_x \lambda_x \alpha(x)$ and use standard properties of absolutely convergent series: if $\alpha(x) = \sum_y \beta_y^x y$, $\sum_y \sum_x \lambda_x \beta_y^x = \sum_x \lambda_x \sum_y \beta_y^x = \sum_x \lambda_x = 1$. In the Kleisli category, outputs are probability distributions. Flipping a coin is a map $\alpha : 1 \rightarrow \{H, T\}$. Let $\beta(H)$ cause two dice to be summed whereas $\beta(T)$ causes one die to be rolled. The Kleisli composition $\beta \circ \alpha : 1 \rightarrow \{1, \dots, 12\}$ performs a correct probability calculation. Noting that $Pr 2$ can be identified with $[0, 1]$, the subobject classifier is the characteristic function of $\{1\}$, $[0, 1] \rightarrow 2$. The box and diamond operators are

$$[\alpha]Q = \{x \in X : \text{if } \alpha(x) = \sum_y \lambda_y y, \text{ then } \lambda_y \neq 0 \Rightarrow y \in Q\}$$

$$\langle \alpha \rangle Q = \{x \in X : \text{if } \alpha(x) = \sum_y \lambda_y y \text{ then there exists } y \in Q \text{ with } \lambda_y \neq 0\}$$

We now paraphrase the Kozen completeness theorem [8] generalized to the setting of Boolean categories [11, Theorem 11.15].

Theorem 2.18 *Let \mathcal{B} be a Boolean category. For X an object of \mathcal{B} , let βX be the set of ultrafilters on the Boolean algebra $\text{Summ}(X)$. Define the **Kozen functor** $K : \mathcal{B} \rightarrow \mathbf{Rel}$ by $KX = \beta X$ and, for $\alpha : X \rightarrow Y$, $\mathcal{U}(K\alpha)\mathcal{V} \Leftrightarrow \forall V \in \mathcal{V} \langle \alpha \rangle V \in \mathcal{U}$. Then K is a taut functor which preserves finite coproducts. By the properties of this functor one shows that any valid sentence of loop-free dynamic logic which is true in \mathbf{Rel} is true in \mathcal{B} . \square*

Remark 2.19 By Proposition 2.13, The Kozen functor for \mathbf{Set}_T is

$$\mathcal{U}(K\alpha)\mathcal{V} \Leftrightarrow \forall V \in \mathcal{V} \{x \in X : V' \notin \text{supp}_Y(\alpha x)\} \in \mathcal{U}$$

Example 2.20 It follows from Theorem 2.7 that any non-trivial submonad of the filter monad \mathbf{F} is taut. Here, the box and diamond operators are given by

$$[\alpha]Q = \{x \in X : Q \in \alpha x\}$$

$$\langle \alpha \rangle Q = \{x \in X : Q' \notin \alpha x\}$$

We now offer a few heuristic remarks concerning proper filters as distributions.

A proper filter \mathcal{F} “homes in” in stages where each $A \in \mathcal{F}$ is a stage. When stages A_1, \dots, A_n have been selected, $A_1 \cap \dots \cap A_n$ is a finer stage. Fixing $\alpha : X \rightarrow FY$, $Q \subset Y$, $[\alpha]Q = \{x \in X : Q \in \alpha x\}$ is the set of all inputs whose outputs allow Q as a stage. The principal ultrafilter $\text{prin}(x)$ is a true point because $\{x\}$ is a stage. In general, the more stages the more a filter can home in so it is natural to consider maximal filters as generalized points. We assume the Boolean prime ideal theorem without further comment, so each (proper) filter is the (nonempty) intersection of the ultrafilters that contain it. The set of such ultrafilters is the set of generalized points the filter homes in on.

A compact Hausdorff space may be regarded as a structure that realizes an actual point for each ultrafilter. Now β is a submonad of \mathbf{F} and it is well known that its algebras are precisely the compact Hausdorff space whose structure map $\beta X \rightarrow X$ is ultrafilter convergence. The forgetful functor from continuous lattices to compact Hausdorff spaces induced by the monad map $\beta \rightarrow \mathbf{F}$ assigns the Lawson topology.

The following proposition supports the single-valuedness of ultrafilters as opposed to filters.

Proposition 2.21 *If G is a submonad of β , the Kozen functor $K : \mathbf{Set}_G \rightarrow \mathbf{Rel}$ factors through \mathbf{Pfn} .*

Proof Let $\alpha : X \rightarrow GY$ and let $\mathcal{U}(K\alpha)\mathcal{V}$, $\mathcal{U}(K\alpha)\mathcal{W}$. If $\mathcal{V} \neq \mathcal{W}$ then, as \mathcal{V}, \mathcal{W} are ultrafilters, $\exists V \in \mathcal{V}$ with $V' \in \mathcal{W}$. Then $\langle \alpha \rangle V = \{x \in X : V' \notin \alpha x\} \in \mathcal{U}$

and $\langle \alpha \rangle V' = \{x \in X : V \notin \alpha x\} \in \mathcal{U}$. As any two elements of \mathcal{U} have non-empty intersection, there exists x with $V, V' \notin \alpha x$, and this is impossible as αx is an ultrafilter. Thus $\mathcal{V} = \mathcal{W}$. \square

While **Rel** is the prototypical Boolean category, the theory of this section suggests that **Set_F** is the prototypical taut-monad Kleisli category. This point of view can be pushed further by independently proving a “Kozen completeness theorem” for **Set_F**. To begin, first observe that for any monad map $\lambda : \mathbf{S} \rightarrow \mathbf{T}$ over any category \mathcal{K} , there is an induced functor $H_\lambda : \mathcal{K}_\mathbf{S} \rightarrow \mathcal{K}_\mathbf{T}$ defined by $H_\lambda(X \xrightarrow{\alpha} SY) = X \xrightarrow{\alpha} SY \xrightarrow{\lambda_Y} TY$. For a taut monad with support map $\text{supp} : \mathbf{T} \rightarrow \mathbf{F}$, $H_{\text{supp}}(X \xrightarrow{\alpha} TY) = X \xrightarrow{\alpha} TY \xrightarrow{\text{supp}_Y} FY$. Comparing the formulas of Proposition 2.13 and Example 2.20, $[\alpha]Q = \{x \in X : Q \in \text{supp}(\alpha x)\} = \{x \in X : Q \in (H_{\text{supp}} \alpha)x\} = [H_{\text{supp}} \alpha]Q$, so that any dynamic-logic assertions which are universally valid in **Set_F** must be universally valid in **Set_T**. In particular, if an assertion holds in **Set_F** it holds in **Rel** = **Set_P**. Applying Theorem 2.18, we have

Theorem 2.22 *Any loop-free assertion of dynamic logic that is valid in **Set_F** is valid in all Boolean categories.* \square

For the most standard collections such as lists, probability distributions, bags, and so forth, a distribution τ has a definite set of members $\text{mem}(\tau)$, e.g. $\text{mem}[a, b, a, c, b] = \{a, b, c\}$, for $\tau = \sum \lambda_x x \in \text{Pr } X$, $\text{mem}(\tau)$ is $\{x : \lambda_x \neq 0\}$. In such cases, $\text{supp}(\tau) = \text{prin}(\text{mem}(\tau))$ is a principal filter ($\text{prin}(A) = \{B : A \subset B\}$). The submonad of principal filters is the subsets monad $\text{prin} : \mathbf{P} \rightarrow \mathbf{F}$. In the principal case, the completeness theorem for **Rel** results via a monad map to **P**. We formalize this definition as follows.

Definition 2.23 A taut monad is **principal** if its support map factors through $\text{prin} : \mathbf{P} \rightarrow \mathbf{F}$. In that case, for $\tau \in TX$, $\text{supp}_X(\tau) = \text{prin}_X(\text{mem}_X(\tau)) = \{A \subset X : \text{mem}_X(\tau) \subset A\}$ which defines the subset $\text{mem}_X(\tau)$ of **members** of τ .

The next lemma will be used in Proposition 5.7. This property was used as an axiom on collections in [12].

Lemma 2.24 *Let **T** be a principal monad, $\alpha : X \rightarrow TY$, $\tau \in TX$. Then*

$$\text{mem}_Y(\alpha^\# \tau) = \bigcup_{x \in \text{mem}_X(\tau)} \text{mem}_Y(\alpha x)$$

Proof Because $\text{mem} : \mathbf{T} \rightarrow \mathbf{P}$ is a monad map, the following square commutes (as reviewed at the beginning of Section 2).

$$\begin{array}{ccc} TX & \xrightarrow{\text{mem}_X} & PX \\ \alpha^\# \downarrow & & \downarrow (\text{mem}_X \alpha)^\# \\ TY & \xrightarrow{\text{mem}_Y} & PY \end{array}$$

Applying the definition of $(\cdot)^\#$ for the power set monad, we have

$$mem_Y(\alpha^\# \tau) = (mem_Y \alpha)^\# mem_X(\tau) = \bigcup_{x \in mem_X(\tau)} mem_Y(\alpha x) \quad \square$$

3 Deterministic Monads

Partial functions are deterministic in that it is not possible to witness two-valued behavior. This may be defined in any category as follows.

Definition 3.1 A morphism $f : X \rightarrow Y$ in any category is **deterministic** if for every coproduct decomposition $Q \rightarrow Y \leftarrow Q'$ of Y there exists a commutative diagram

$$\begin{array}{ccccc} P & \longrightarrow & X & \longleftarrow & P' \\ \downarrow & & \downarrow f & & \downarrow \\ Q & \longrightarrow & Y & \longleftarrow & Q' \end{array}$$

in which the top row is a coproduct decomposition of X . If, addition, this can always be done with both squares pullbacks, we say f is **crisp**.

This definition does not require any coproducts or pullbacks to exist. For the Kleisli category of any monad in **Set**, coproducts are disjoint unions which meshes well with the philosophy behind the definition.

Definition 3.2 A monad is **deterministic** if all morphisms in its Kleisli category are deterministic. Similarly, a monad is **crisp** if all morphisms in its Kleisli category are crisp.

Example 3.3 Let $\beta_\star X = \{\mathcal{U} \in FX : \mathcal{U} \text{ is an ultrafilter or } \mathcal{U} = 2^X\}$. We check that β_\star is a submonad of F . We have $prin(x) \in \beta X \subset \beta_\star X$. Now let $\alpha : X \rightarrow \beta_\star Y$, $\mathcal{U} \in \beta_\star X$. We must show that $\alpha^\# \mathcal{U} = \{B \subset Y : \{x \in X : B \in \alpha x\} \in \mathcal{U}\} \in \beta_\star Y$. Clearly $\mathcal{U} = 2^X \Rightarrow \alpha^\# \mathcal{U} = 2^Y$. Otherwise, $\mathcal{U} \in \beta X$. Let $B \subset Y$. To show: $B \in \alpha^\# \mathcal{U}$ or $B' \in \alpha^\# \mathcal{U}$ (possibly both). Suppose $B \notin \alpha^\# \mathcal{U}$. As \mathcal{U} is an ultrafilter, $U = \{x : B \notin \alpha x\} \in \mathcal{U}$. For $x \in U$, $\alpha x \neq 2^Y$ so $\alpha x \in \beta Y$ and $\{x : B' \in \alpha x\} = U \in \mathcal{U}$, so $B' \in \alpha^\# \mathcal{U}$ as desired.

It is easy to check that the β_\star -algebras are compact Hausdorff spaces with base point and that the discrete space 2 with base point *True* is the support classifier.

Definition 3.4 In a Boolean category, $f : X \rightarrow Y$ is **total** if $[f]0 = 0$.

Lemma 3.5 [11, Proposition 12.2, Corollary 12.3] For $f : X \rightarrow Y$ in a Boolean category, the following hold.

- (1) f is total $\Leftrightarrow \forall Q \quad \langle \alpha \rangle Q \supset [\alpha]Q$.
- (2) f is deterministic $\Leftrightarrow \forall Q \quad \langle \alpha \rangle Q \subset [\alpha]Q$.
- (3) f is crisp $\Leftrightarrow \forall Q \quad \langle \alpha \rangle Q = [\alpha]Q$.

(4) f is crisp $\Leftrightarrow Q \mapsto [f]Q$ is a Boolean algebra homomorphism $\text{Summ}(Y) \rightarrow \text{Summ}(X)$. \square

Thus a map in a Boolean category is crisp if and only if it is total and deterministic.

By [13, Proposition 3.12] any submonad G of the filter monad \mathbf{F} is taut and the inclusion $\iota : \mathbf{G} \rightarrow \mathbf{F}$ is a taut monad map.

For $f : X \rightarrow Y$ a function and \mathbf{T} a taut monad, $f^\diamond = X \xrightarrow{f} Y \xrightarrow{\eta_Y} TY$ is crisp in $\mathbf{Set}_{\mathbf{T}}$. the converse is far from true, as we see in the next proposition, which provides further evidence that ultrafilters are like points.

Proposition 3.6 *Every submonad of β is crisp.*

Proof Let G be a submonad of β . For $\alpha : X \rightarrow GY$, $[\alpha]Q = \{x \in X : Q \in \alpha x\}$ and $\langle \alpha \rangle Q = \{x \in X : Q' \notin \alpha x\} = \{x \in X : Q \in \alpha x\}$ as αx is an ultrafilter. Thus $[\alpha]Q = \langle \alpha \rangle Q$ so α is crisp. \square

We can now relate deterministic maps to β_* .

Proposition 3.7 *For a taut monad \mathbf{T} , \mathbf{T} is crisp if and only if $\text{supp} : \mathbf{T} \rightarrow \mathbf{F}$ factors through β . \mathbf{T} is deterministic if and only if supp factors through β_* .*

Proof For $\alpha : X \rightarrow TY$ is crisp $\Leftrightarrow \forall Q [\alpha]Q = \langle \alpha \rangle Q$. By Proposition 2.13 this happens if and only if $(\forall x \forall Q Q \in \text{supp}(\alpha x) \Leftrightarrow Q' \notin \text{supp}(\alpha x)) \Leftrightarrow \forall x \alpha x \in \beta X$. As any $\tau \in TX$ has form αx for some α , this proves the first statement. Similarly, α is deterministic $\Leftrightarrow (\forall x \forall Q Q' \notin \text{supp}(\alpha x) \Rightarrow Q \in \text{supp}(\alpha x)) \Leftrightarrow \forall x \alpha x \in \beta_* X$. \square

Corollary 3.8 *Any taut monad \mathbf{T} contains a largest crisp submonad T_{crisp} whose Kleisli category consists of the crisp morphisms of $\mathbf{Set}_{\mathbf{T}}$. Similarly, \mathbf{T} has a largest deterministic submonad T_{det} whose Kleisli category consists of the deterministic morphisms of $\mathbf{Set}_{\mathbf{T}}$.*

Proof Form the pointwise pullbacks (which are pullbacks in the category of monads and monad maps)

$$\begin{array}{ccccc} T_{\text{crisp}} & \longrightarrow & T_{\text{det}} & \longrightarrow & T \\ \downarrow & & \downarrow & & \downarrow \text{supp} \\ \beta & \longrightarrow & \beta_* & \longrightarrow & F \end{array}$$

and then use the Proposition. (All monad maps in the diagram are taut). \square

Lemma 3.9 *Let \mathbf{S} , \mathbf{T} be taut monads, $\lambda : \mathbf{S} \rightarrow \mathbf{T}$ a taut monad map. Then if \mathbf{T} is deterministic so, too, is \mathbf{S} ; if \mathbf{T} is crisp then \mathbf{S} is also.*

Proof For $\alpha : X \rightarrow SY$, $Q \subset Y$ with inclusions $j : Q \rightarrow Y$, $j' : Q' \rightarrow Y$, consider the diagram whose explanation follows:

$$\begin{array}{ccccc}
P & \xrightarrow{i} & X & \xleftarrow{i'} & P' \\
\beta \downarrow & & \downarrow \alpha & & \downarrow \beta' \\
SQ & \xrightarrow{Sj} & SY & \xleftarrow{Sj'} & SQ' \\
\lambda_Q \downarrow & & \downarrow \lambda_Y & & \downarrow \lambda_{Q'} \\
TQ & \xrightarrow{Tj} & TY & \xleftarrow{Tj'} & TQ'
\end{array}$$

The bottom squares are the naturality squares for λ induced by the monics j, j' and so these are pullbacks. As $\lambda_Y \alpha$ is deterministic in $\mathbf{Set}_{\mathbf{T}}$ by hypothesis, there exists $P \subset X$ with inclusions $i : P \rightarrow X$, $i' : P' \rightarrow X$ and maps $P \rightarrow TQ$, $P' \rightarrow TQ'$ completing the perimeter of the diagram. By the pullback property, these maps factor via β, β' through SQ and SQ' , completing the diagram. Thus α is deterministic. If $\lambda_Y \alpha$ is crisp as well then, from the same diagram, using elementary pullback-pasting and the fact that S preserves monics, α is crisp. \square

Theorem 3.10 *Let \mathbf{T} be a taut monad in \mathbf{Set} . Then \mathbf{T} is deterministic if and only if there exists a taut monad map $\mathbf{T} \rightarrow \beta_*$. \mathbf{T} is crisp if and only if there exists a taut monad map $\mathbf{T} \rightarrow \beta$.*

Proof We give the proof for “deterministic”, the crisp case being entirely similar. If \mathbf{T} is deterministic, by Proposition 3.7, $\text{supp} = \mathbf{T} \xrightarrow{\lambda} \beta_* \xrightarrow{\iota} \mathbf{F}$ with ι the submonad inclusion. As ι and $\iota \lambda$ are taut monad maps, so is λ . Conversely, suppose $\lambda : \mathbf{T} \rightarrow \beta_*$ is a taut monad map. The taut inclusion $\iota : \beta_* \rightarrow \mathbf{F}$ is its own support map by [13, Proposition 3.6] so that β_* is deterministic by Proposition 3.7. Thus \mathbf{T} is deterministic by Lemma 3.9. \square

We emphasize that \mathbf{T} is necessarily a taut monad if it admits a taut monad map to a taut monad, so the a priori assumption that \mathbf{T} be taut in the previous theorem is necessary only in one direction.

Example 3.11 The exception monad \mathbf{Ex} of Example 2.14 is taut and deterministic since

$$X + \star \xrightarrow{\text{prin}_X + \text{id}} \beta_* X$$

is a taut monad map. (This holds, as well, for any set E of exceptions in that example).

Theorem 3.12 *Let \mathbf{T} be a principal taut monad. Then \mathbf{T} is deterministic if and only if \mathbf{T} admits a taut monad map to the exception monad.*

Proof If $\text{prin}(A)$ is an ultrafilter, A is a singleton so $\beta_* X \cap PX = \mathbf{Ex} X$. \square

At this point one might wonder if *every* deterministic taut monad admits a taut monad map to the exception monad. The next example shows that this is not the case.

Example 3.13 There is no monad map $\beta_* \rightarrow \mathbf{Ex}$. To see this, note that the variety of \mathbf{Ex} -algebras is the category \mathbf{Set}_* of sets with base point whereas, as has already been discussed above, the variety of β_* -algebras is the category $\mathbf{CT2}_*$ of

compact Hausdorff spaces with base point. It must be shown, then, that there is no forgetful functor $H(X, *) = (X, *, \mathcal{T})$ with \mathcal{T} a compact Hausdorff topology on X in such a way that every function $f : (X, *) \rightarrow (Y, \star)$ with $f(*) = \star$ is such that $f : H(X, *) \rightarrow H(Y, \star)$ is continuous. To find a contradiction, suppose that such H exists and choose $X = \{x_1, x_2, \dots\}$ with base point x_1 and write $H(X, x_1) = (X, x_1, \mathcal{T})$. By Hausdorff, there exists $\emptyset \neq U \in \mathcal{T}$ with $x_1 \notin U$. For any $\emptyset \neq A \subset X$ with $x_1 \notin A$, let $f : X \rightarrow X$ be any function with $f(x) = x_1$ if $x \notin A$ whereas $f(A) \subset U$. Then $A = f^{-1}(U)$ is open. This shows that $X \setminus \{x_1\}$ is an open discrete subspace. Now let $x_1 \in V \in \mathcal{T}$. As $\{V\} \cup \{\{x\} : x \notin V\}$ has a finite subcover, $X \setminus V$ is finite. Thus if \mathcal{S} is the one-point-compactification topology induced by adding x_1 to the locally compact discrete space $X \setminus \{x_1\}$, $\mathcal{T} \subset \mathcal{S}$. As both topologies are compact Hausdorff, $\mathcal{T} = \mathcal{S}$. Now let $g : X \rightarrow X$ be defined by $g(x_1) = x_1$, $g(x) = x_2$ if $x \neq x_1$. As $\lim(x_n : n = 0, 1, \dots) = x_1$, g is not continuous. This is the desired contradiction.

Example 3.14 $\mathbf{Rel}_{\text{det}} = \mathbf{Pfn}$. For the submonad \mathbf{P}_0 of the subsets monad \mathbf{P} , $P_0X = \{A \subset X : A \text{ is finite}\}$, The Kleisli category is the category of sets and finite-valued relations. The deterministic maps also produce the category \mathbf{Pfn} .

Example 3.15 Let $\mathbf{L} = (L, \eta, \mu)$ be the list monad. The deterministic maps $f : X \rightarrow LY$ in $\mathbf{Set}_{\mathbf{L}}$ satisfy $fx = y^n$ for some $y \in Y$, $n \geq 0$ because \mathbf{L}_d arises as the pullback

$$\begin{array}{ccc} \mathbf{L}_d & \longrightarrow & \mathbf{Ex} \\ \downarrow & & \downarrow \\ \mathbf{L} & \xrightarrow{\text{mem}} & \mathbf{P}_0 \end{array}$$

where $EX = X + \star \rightarrow P_0X$ maps x to $\{x\}$ and the unique element of \star to \emptyset . Noting that $y^0 = z^0$ even if $y \neq z$, let $\mathbb{N}_+ = \{1, 2, 3, \dots\}$, and define $L_dX = (X \times \mathbb{N}_+) + \{\perp\}$. In the Kleisli category, maps are partial functions whose values include an integer ≥ 1 and these integers multiply when maps are composed.

Example 3.16 Let M be a monoid with unit e and multiplication mn . Let

$$\begin{aligned} TX &= (X + \{\perp\}) \times M \\ X &\xrightarrow{\eta_X} TX, \quad x \mapsto (x, e) \\ TTX &\xrightarrow{\mu_X} TX, \quad (\perp, m) \mapsto (\perp, m), \quad ((x, m), n) \mapsto (x, mn) \end{aligned}$$

Such \mathbf{T} is a deterministic taut monad. When M is $(\mathbb{N}, +, 0)$, the morphisms may be interpreted as partial functions whose values return a computation time.

Example 3.17 It is obvious that $\mathbf{F}_{\text{det}} = \beta_*$. It is easily checked that $NX = \{\mathcal{F} \in FX : \bigcap \mathcal{F} \neq \emptyset\}$ is a submonad of \mathbf{F} . We called this the **neighborhood monad** in [10] since the most well known example of elements of NX are neighborhood filters in a topological space. As $NX \cap \beta_*X$ is the set of principal ultrafilters on X , $\mathbf{N}_{\text{det}} = \mathbf{N} \cap \beta_*$ is the identity monad. Thus the only deterministic maps in the Kleisli category of \mathbf{N} are the ordinary total functions between sets.

The **N**-algebras are “continuous lattices without top”. A more precise description is as follows. An *N*-**semilattice** is a partially ordered set with all non-empty infima (and hence all bounded suprema) such that for every non-empty family $(D_i : i \in I)$ with each D_i a bounded directed subset of X , the following equation holds:

$$\bigwedge_i \left(\bigvee D_i \right) = \bigvee_{d \in \prod_i D_i} \left(\bigwedge_i d_i \right)$$

We leave it to the reader to verify that **Set**^{**N**} is the category of all *N*-semilattices with morphisms that preserve non-empty infima and bounded directed suprema. (Hint: show that (NX, \subset) is the free *N*-semilattice generated by X with $f^\# : NX \rightarrow Y$ defined by $f^\#(\mathcal{F}) = \bigvee_{A \in \mathcal{F}} \bigwedge (fA)$).

4 Comparison with Restriction Categories

Definition 4.1 A **restriction category** [4] is a category equipped with a unary operation $f : X \rightarrow Y \mapsto \bar{f} : X \rightarrow X$ subject to the following four axioms:

- (R.1) For $f : X \rightarrow Y$, $f\bar{f} = f$.
- (R.2) For $f : X \rightarrow Y$, $g : X \rightarrow Z$, $\bar{f}\bar{g} = \bar{g}\bar{f}$.
- (R.3) For $f : X \rightarrow Y$, $g : X \rightarrow Z$, $\overline{g\bar{f}} = \bar{g}\bar{f}$.
- (R.4) For $X \xrightarrow{f} Y \xrightarrow{g} Z$, $\bar{g}f = f\bar{g}\bar{f}$.

In any restriction category, each endomorphism of form $\bar{f} : X \rightarrow X$ is an idempotent. We call it a **restriction idempotent** and the set $R(X)$ of all such restriction idempotents is a commutative idempotent semigroup and so forms a semilattice. The interpretation of \bar{f} is “the domain of definition of f ”, e.g. $\bar{f}x = x$ if fx is defined and is otherwise undefined. It is discussed in precise detail in [5] that the axioms are not compatible with “more than one exception”, so our discussion here is only intuitive. In any restriction category, a morphism f is **total** if $\bar{f} = id$. Any category is a restriction category if all maps are declared total, so that there need not be any exceptions. The interpretation of Axiom (R.4) is that all maps are deterministic. To see the idea, suppose fx includes two distinct outputs a, b . Let ga be defined, gb not. Then b is an output of $f\bar{g}\bar{f}x = fx$ whereas $\bar{g}f x = a$ so (R.4) fails.

Let us be more precise for **Set**_{**T**} with **T** taut. View $T\emptyset$ (the distributions on the empty set) as the set of exceptions of **T**. The condition that there is exactly one exception is then that $T\emptyset = \star$ and we call **T** a **zero monad** if it satisfies this condition since then \emptyset is the zero object of the Kleisli category. This Kleisli category then has a natural definition of restriction as follows:

Definition 4.2 For $\alpha : X \rightarrow TY$ with **T** a zero monad, define $\bar{\alpha} : X \rightarrow TX$ by the following diagram in which the top row is a coproduct:

$$\begin{array}{ccccc}
[\alpha]\emptyset & \xrightarrow{i'} & X & \xleftarrow{i} & ([\alpha]\emptyset)' \\
\downarrow & & \downarrow \bar{\alpha} & & \downarrow i \\
T\emptyset & \xrightarrow{Tj'} & TX & \xleftarrow{\eta_X} & X
\end{array}$$

where i', i, j are inclusions.

Theorem 4.3 [15, Theorem 6.4] *For a taut zero monad \mathbf{T} with the restriction operator of Definition 4.2, \mathbf{Set}_T is a restriction category if and only if \mathbf{T} is a deterministic monad.* \square

Example 4.4 The monad β_* is a deterministic zero monad. \mathbf{F} is a zero monad which is not deterministic. β and \mathbf{N} are not zero monads.

The intense study of domains in computation theory begins by observing that the set $\mathbf{Pfn}(X, Y)$ of partial functions from X to Y is ordered by extension. This is available in any restriction category as follows.

Every restriction category is an ordered category. For $f, g : X \rightarrow Y$ say that $f \leq g$ if $g\bar{f} = f$, literally “ g restricted to the domain of f is f ”. This is always a partial order and composition on either side is monotone.

The study of domains has for the most part not focused on the more mundane programming constructs such as if-then-else. But this construct is readily available in the ordered category \mathbf{Pfn} : if $P \subset X$ with inclusions $i : P \rightarrow X$, $i' : P' \rightarrow X$, and $f, g : X \rightarrow Y$, if P then f else $g : X \rightarrow Y$ is $f\bar{i} \vee gi'$. What happens here, is that the semilattice $R(X)$ of restriction idempotents is 2^X and this is the Boolean algebra of tests. A general class of ordered sets that have the necessary structure was introduced in [5], and are defined next.

Definition 4.5 A partially ordered set P with a least element 0 is a **locally Boolean poset** if for all $x \in P$ the principal downset $\downarrow x = [0, x]$ is a Boolean algebra whose finite suprema are suprema in P . If P, Q are locally Boolean posets, a function $f : P \rightarrow Q$ is a **locally Boolean map** if f is monotone, $f(0) = 0$ and, for all x , the induced function $[0, x] \rightarrow [0, fx]$ is a Boolean algebra homomorphism. Denote the category of locally Boolean posets as **LBP**.

A natural associated class of restriction categories is then as follows.

Definition 4.6 A **classical restriction category** is a restriction category \mathcal{C} which satisfies the following two axioms:

- (CR.1) \mathcal{C} is enriched over **LBP**, that is, the homset $\mathcal{C}(X, Y)$ under the restriction ordering $f \leq g$ if $g\bar{f} = f$ is a locally Boolean poset and composition on either side is a locally Boolean map.
- (CR.2) Denoting the least element of $\mathcal{C}(X, Y)$ as $0 : X \rightarrow Y$, if $f, g : X \rightarrow Y$ are **disjoint**, written $f \perp g$, in that $\bar{f}\bar{g} = 0$ then $f \vee g$ exists in $\mathcal{C}(X, Y)$.

We regard Boolean algebras as forming a variety of universal algebras. As such, 0 is the terminal object so $[0, x] \rightarrow [0, 0]$ is a Boolean algebra homomorphism.

Example 4.7 \mathbf{Pfn} is a classical restriction category. Here, $\bar{f}(x) = x$ if fx is defined and is otherwise undefined. $[0, f] \cong 2^D$ with $D = \{x : fx \text{ is defined}\}$ is how homsets are locally Boolean posets. A somewhat more gradual explanation can be given as follows. In \mathbf{LBP} , coproducts exist as the disjoint union poset with all least elements identified. Products exist in \mathbf{LBP} using the pointwise-ordered product poset. Every Boolean algebra is a locally Boolean poset. Start with the Boolean algebra 2 . The flat poset A_\perp obtained by adding a bottom element to discrete A is simply a typical copower of 2 in \mathbf{LBP} . The locally Boolean poset power $(Y_\perp)^X$ is precisely the poset $\mathbf{Pfn}(X, Y)$. The local Booleanness of composing on either side now follows from standard properties of product and coproduct.

\mathbf{Ex} is a deterministic zero monad and its Kleisli category (namely \mathbf{Pfn}) is a classical restriction category. A surprise of this paper is that this happens for every deterministic zero monad.

Theorem 4.8 *The Kleisli category of a deterministic zero monad $\mathbf{T} = (T, \eta, \mu)$ is a classical restriction category.*

Proof $\mathbf{Set}_\mathbf{T}$ is a Boolean category with zero maps and all maps deterministic, and is a restriction category as in Definition 4.2 by Theorem 4.3. That such a category is a classical restriction category is one of the main theorems of [5]. We briefly indicate how the locally Boolean structure arises in this specific context. The lattice structure of $[0, \gamma]$ is as follows. If $\alpha, \beta, \gamma : X \rightarrow Y$ with $\alpha, \beta \leq \gamma$. The infimum $\alpha \wedge \beta$ is $\alpha \bar{\beta}$. The supremum $\alpha \vee \beta$ is defined by

$$\begin{array}{ccccc} [\alpha]\emptyset & \xrightarrow{i} & X & \xleftarrow{j} & ([\alpha]\emptyset)' \\ i^\diamond \downarrow & & \downarrow \alpha \vee \beta & & \downarrow j^\diamond \\ TX & \xrightarrow{\alpha} & TY & \xleftarrow{\beta} & TX \end{array}$$

where i, j are inclusions. Now, the restriction idempotents of X constitute the interval $[0, 1]$ which is order-isomorphic to $\text{Summ}(X)$ by [11, Theorem 8.15] and so forms a Boolean algebra. Thus $[0, \bar{\gamma}]$ is an interval in a Boolean algebra so is itself a Boolean algebra. It is shown that $[0, \gamma] \rightarrow [0, \bar{\gamma}]$, $\alpha \mapsto \bar{\alpha}$ is an order isomorphism. \square

5 Dynamic Logic of Spaces

In this section, we observe that for any topological space X there is an appropriate endomorphism α in a taut-monad Kleisli category for which $[\alpha]Q$, $\langle \alpha \rangle Q$ are, respectively, the interior and closure operators of the space. Thus any valid formula of dynamic logic involving one α is a law of topological spaces. We also study classes of topological spaces induced by taut monads. We begin with a general definition and proposition for taut monads.

Definition 5.1 Let \mathbf{T} be a taut monad. A **\mathbf{T} -topological space** is a pair (X, t) with $t : X \rightarrow TX$ an idempotent in $\mathbf{Set}_\mathbf{T}$ such that $[t]Q \subset Q$ for all $Q \subset X$. A subset Q is **t -open** if $[t]Q = Q$.

Proposition 5.2 *For a \mathbf{T} -topological space (X, t) , the t -open sets form a topology with interior operator $[t]Q$ and closure operator $\langle t \rangle Q$.*

Proof \emptyset is open since $[t]\emptyset \subset \emptyset$. By elementary facts about pullbacks, $[t]X = X$ and $[t](Q \cap R) = [t]Q \cap [t]R$ so X is t -open and a finite intersection of t -opens is t -open. If $V = \bigcup V_i$ with each V_i t -open, $x \in V \Rightarrow \exists i \ x \in V_i = [t]V_i \subset [t]V$ so V is t -open. So far, the t -opens form a topology. For any $Q \subset X$, $[t]Q$ is t -open since (in any Boolean category) $[\alpha](\beta Q) = [\beta\alpha]Q$ and t is idempotent with $[t]Q \subset Q$. If $x \in Q^\circ$ there exists $U = [t]U$ with $x \in U \subset Q$. Thus $x \in [t]U \subset [t]Q$. This shows $[t]Q = Q^\circ$. But then $\overline{Q} = Q'^{\circ'} = ([t]Q')' = \langle t \rangle Q$. \square

Theorem 5.3 *Every topological space is an \mathbf{F} -topological space for the filter monad \mathbf{F} .*

Proof Let X be a topological space with neighborhood filter \mathcal{N}_x at x . Define $t : X \rightarrow FX$ by $tx = \mathcal{N}_x$. Then $t^\# : FX \rightarrow FX$ maps \mathcal{F} to $\{A \subset X : \{x \in X : A \in \mathcal{N}_x\} \in \mathcal{F}\} = \{A \subset X : A^\circ \in \mathcal{F}\}$ so that $t^\#(tx) = t^\#(\mathcal{N}_x) = \{A \subset X : A^\circ \in \mathcal{N}_x\} = \mathcal{N}_x$, which shows t is idempotent. By Example 2.20, $[t]Q = \{x \in X : Q \in tx\} = \{x \in X : Q \in \mathcal{N}_x\} = Q^\circ$ so $[t]Q \subset Q$. \square

Example 5.4 Segerburg's axiom [20] is

$$[\alpha]Q \cap \langle \alpha \rangle R \subset \langle \alpha \rangle (Q \cap R)$$

This example is obviously true in \mathbf{Rel} , so by the Kozen completeness theorem it must hold in all Boolean categories. Hence any topological space must satisfy

$$Q^\circ \cap \overline{R} \subset \overline{Q \cap R}$$

Example 5.5 In \mathbf{Rel} , $\alpha : X \rightarrow PY$ is a partial function if and only if for all subsets Q, R of Y , $\langle \alpha \rangle (Q \cup R) = \langle \alpha \rangle Q \cup \langle \alpha \rangle R$. Equivalently, α is a partial function if and only if $\langle \alpha \rangle Q \subset [\alpha]Q$ for all Q . It follows that a topological space is discrete if and only if $(Q \cup R)^\circ = Q^\circ \cup R^\circ$ since $\overline{Q} \subset Q^\circ$ obviously forces discreteness.

It is clear that Theorem 5.3 can be proved using any submonad of \mathbf{F} which contains all neighborhood filters, such as \mathbf{N} . However, no submonads of β_* will yield interesting spaces, as the following example attests.

Example 5.6 If \mathbf{T} is a deterministic monad then every \mathbf{T} -topological space is discrete. This follows from Lemma 3.5 (2) since any \mathbf{T} -topological space satisfies $\overline{Q} \subset Q^\circ$.

A topological space is an **Alexandroff space** [1] if every intersection of open sets is open. It is well known that the category of Alexandroff spaces and continuous maps is isomorphic to the category of preordered sets (sets with reflexive and transitive relation) and monotone maps. The open sets are the lower sets, so $\downarrow x$ is the minimum neighborhood of x , and the closed sets are the upper sets. Here, $x \leq y \Leftrightarrow y \in \overline{\{x\}}$.

Proposition 5.7 *If \mathbf{T} is a principal monad then every \mathbf{T} -topological space is Alexandroff.*

Proof Let (X, t) be a \mathbf{T} -topological space, so that $t \circ t = t$, $[t]Q \subset Q$. We have

$$[t]Q = \{x \in X : Q \in \text{supp}(tx)\} = \{x \in X : \text{mem}(tx) \subset Q\}$$

so Q is open if and only if $\text{mem}(tx) \subset Q \Rightarrow x \in Q$. Write $x \leq y$ if $y \in \text{mem}(tx)$. As $\text{mem}(tx) \subset \text{mem}(tx)$, $x \in [t](\text{mem}(tx) \subset \text{mem}(tx))$, so $x \leq x$. By Lemma 2.24,

$$\text{mem}(tx) = \text{mem}(t^\#(tx)) = \bigcup_{y \in \text{mem}(tx)} \text{mem}(ty)$$

so $y \in \text{mem}(tx) \Rightarrow \text{mem}(ty) \subset \text{mem}(tx)$. Thus if $x \leq y$, $y \leq z$ then $y \in \text{mem}(tx)$, $z \in \text{mem}(ty) \subset \text{mem}(tx)$ and $x \leq z$. This shows that \leq is a preorder. If $x \in Q^\circ = [t]Q$ and $x \leq y$ then $y \in \text{mem}(tx) \subset Q \Rightarrow \text{mem}(ty) \subset \text{mem}(tx) \subset Q$, so $y \in Q^\circ$. Thus Q° is an upper set. Conversely, if Q is an upper set and $x \in Q$, $y \in \text{mem}(tx)$ then $x \leq y \Rightarrow y \in Q$ so $\text{mem}(tx) \subset Q$. This shows that $Q \subset [t]Q$. We have shown that the open sets are the upper sets of a preorder, so the space is Alexandroff. \square

Example 5.8 For \mathbf{P} the power set monad, the \mathbf{P} -topological spaces are precisely the Alexandroff spaces.

To see this, since \mathbf{P} is a principal monad, the previous proposition gives that all its topological spaces are Alexandroff. Conversely, if $t : X \rightarrow PX$ is any reflexive and transitive relation then $t \circ t$ (being the composition in \mathbf{Rel}) $\subset t$ whereas $t = t \circ \text{id}_X \subset t \circ t$, so t is idempotent. If $tx \subset Q$ then $x \in tx \subset Q$ so $[t]Q \subset Q$. Thus a reflexive and transitive relation is a \mathbf{P} -topology. As $tx = \uparrow x$, $Q^\circ = [t]Q = \{x : \uparrow x \subset Q\}$, the open sets are the upper sets. Hence an arbitrary Alexandroff space is a \mathbf{P} -topological space. \square

The nicest results for \mathbf{G} -topological spaces arise when G is a submonad of the filter monad and we conclude with some results of that type, beginning with

Proposition 5.9 *Let G be a submonad of the filter monad \mathbf{F} . Then the following hold.*

- (1) *If (X, t) is a \mathbf{G} -topological space, $\forall x \in X$, tx is the neighborhood filter \mathcal{N}_x of x .*
- (2) *A space X arises as a \mathbf{G} -topological space if and only if $\forall x \in X$, $\mathcal{N}_x \in GX$.*

Proof Let (X, t) be a \mathbf{G} -topological space. By Example 2.20, $Q^\circ = [t]Q = \{x \in X : Q \in tx\}$ so that $\mathcal{N}_x = \{Q \subset X : x \in Q^\circ\} = \{Q \subset X : Q \in tx\} = tx$. This proves the first statement from which the second is clear, given the proof of Theorem 5.3. \square

Despite the uselessness of submonads of β for producing spaces, as demonstrated by Example 5.6, the next proposition provides a construction which transforms arbitrary submonads of \mathbf{F} (including of β) into useful submonads.

Proposition 5.10 *Let G be a submonad of \mathbf{F} . Then*

$$\widehat{GX} = \{\mathcal{F} \in FX : \mathcal{F} = \bigcap (\mathcal{G}_i : i \in I) \text{ with } I \neq \emptyset, \mathcal{G}_i \in GX\}$$

is again a submonad of \mathbf{F} .

Proof For $x \in X$, $\text{prin}(x) \in GX \subset \widehat{GX}$. Now let $\alpha : X \rightarrow \widehat{GY}$, $\mathcal{F} \in \widehat{GX}$ and write $\mathcal{F} = \bigcap \mathcal{G}_i$ with $\mathcal{G}_i \in GX$. then $B \in \alpha^\# \mathcal{F} \Leftrightarrow \{x : B \in \alpha x\} \in \mathcal{F} \Leftrightarrow \forall i \{x : B \in \alpha x\} \in \mathcal{G}_i \Leftrightarrow \forall i B \in \alpha^\# \mathcal{G}_i$. As $\alpha^\# \mathcal{G}_i \in GY$, $\alpha^\# \mathcal{F} = \bigcap_i \alpha^\# \mathcal{G}_i \in \widehat{GY}$. \square

Recall [17] that a space **has countable tightness** if whenever $x \in \overline{A}$ there exists a countable subset $C \subset A$ with $x \in \overline{C}$.

Proposition 5.11 *Let $\beta_\omega X = \{\mathcal{U} \in \beta X : \text{some member of } \mathcal{U} \text{ is countable}\}$. Then β_ω is a submonad of β and every $\widehat{\beta}_\omega$ -topological space has countable tightness.*

Proof For $x \in X$, $\text{prin}(x) \in \widehat{\beta}_\omega X$ since $\{x\}$ is countable. Now let $\alpha : X \rightarrow \beta_\omega Y$, $\mathcal{U} \in \beta_\omega X$. Let A be a countable member of \mathcal{U} and let B_x be a countable member of αx . Then $C = \bigcup_{x \in A} B_x$ is countable. For $x \in A$, $B_x \in \alpha x \Rightarrow C \in \alpha x$. Thus $\{x \in X : C \in \alpha x\} \supset A$ so A is a countable member of $\alpha^\# \mathcal{U}$. This proves β_ω is a submonad. Now let (X, t) be a $\widehat{\beta}_\omega$ -topological space. Applying Proposition 5.9, every neighborhood filter is in $\widehat{\beta}_\omega X$. Let $x \in \overline{A}$. Then $A' \notin \mathcal{N}_x$ so there exists $\mathcal{V} \in \beta_\omega X$ with $\mathcal{N}_x \cup \{A\} \subset \mathcal{V}$. Let C be a countable member of \mathcal{V} . Then $A \cap C$ is a countable subset of A . As $A \cap C \in \mathcal{V}$, every neighborhood of x intersects it, so $x \in \overline{A \cap C}$. \square

References

- [1] P. S. Alexandroff, *Über die Metrisation der im kleinen kompakten topologische Räume*, Mathematische Annalen 92, 1924, 294–301.
- [2] Barr, M., *Coequalizers and free triples*, Mathematische Zeitschrift 116, 1970, 307–322.
- [3] Bird, R. and de Moor, O., *Algebra of Programming*, Prentice-Hall, Europe, 1998.
- [4] Cockett, J. R. B. and Lack, S., *Restriction Categories I: Categories of partial maps*, Theoretical Computer Science 270 (2002), 223–259.
- [5] Cockett, J. R. B. and Manes, E. G., *From Boolean categories to classical restriction categories*, to appear.
- [6] Day, A., *Filter monads, continuous lattices and closure systems*, Canadian Journal of Mathematics XXVII, 1975, 50–59.
- [7] Fischer, M. J., and Ladner, R. E., *Propositional dynamic logic of regular programs*, Journal of Computer and System Sciences 18, 1979, 282–294.
- [8] Kozen, D., *A representation theorem for models of *-free PDL*, in J. W. de Bakker and J. Van Leeuwen (eds.), Automata, Languages and Programming, ICALP '80, Lecture Notes in Computer Science 85, Springer, 1980, 351–362.
- [9] Mac Lane, S., *Categories for the Working Mathematician*, Springer, Berlin, 1971.
- [10] Manes, E. G., *A class of fuzzy theories*, Journal of Mathematical Analysis and Applications 85, 1982, 409–451.
- [11] Manes, E. G., *Predicate Transformer Semantics*, Cambridge University Press, 1992.
- [12] Manes, E. G., *Implementing collection classes with monads*, Mathematical Structures in Computer Science 8 (1998), 231–276.

- [13] Manes, E. G., *Taut monads and T0 spaces*, Theoretical Computer Science 275 (2002), 79–109.
- [14] Manes, E. G., *Monads of Sets*, in M. Hazewinkel (ed.), Handbook of Algebra, Vol. 3, Elsevier, Amsterdam, 2003, 67–153.
- [15] Manes, E. G., *Boolean restriction categories and taut monads*, Theoretical Computer Science, 360 (2006), 77–95.
- [16] Moggi, E., *Notions of computation and monads*, Information and Computation 93 (1991), 55–92.
- [17] Moore, R. C. and Mrówka, S., *Topologies determined by countable objects*, Notices of the American Mathematical Society 11, 1964, 554.
- [18] Pratt, V. R., *Semantical considerations on Floyd-Hoare logic*, Proceedings of the 17th IEEE Symposium on the Foundations of Computer Science, 1976, 109–121.
- [19] Riesz, F., *Stetigkeitsbegriff und abstrakte mengenlehre*, Proceedings of the International Congress of Mathematicians, Series I, Fourth Congress, Rome, 2 (1908), 18–24; Kraus Reprint Limited, Nendeln/Liechtenstein, 1967.
- [20] Segerburg, K., *A completeness theorem in the modal logic of programs*, Notices of the American Mathematical Society 24, 1977, A-522.
- [21] Wadler, P., *Comprehending monads*, Mathematical Structures in Computer Science 2 (1992), 461–493.
- [22] Wyler, O., *Algebraic theories of continuous lattices*, Lecture Notes in Mathematics 871, Springer, Berlin, 1981, 390–413.