Available online at www.sciencedirect.com
SCIENCE DIRECT®

Theoretical Computer Science 360 (2006) 77–95

Theoretical Computer Science

www.elsevier.com/locate/tcs

Boolean restriction categories and taut monads

Ernie Manes*

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

Received 28 October 2004; received in revised form 21 November 2005; accepted 9 February 2006

Communicated by B.P.F. Jacobs

Abstract

A Boolean category is a restriction category if and only if it has one exception and all morphisms are deterministic. In the category of sets, taut monads are precisely the Boolean ones. It follows that collection monad types in Haskell inherit an assertion calculus based on dynamic logic.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Restriction categories; Boolean categories; Monads in programming; Dynamic logic

1. Introduction

Three category-theoretic constructs in programming semantics are *Boolean categories*, *restriction categories* and *taut monads*. We begin by saying enough about these to establish their relevance to theoretical computer scientists.

The Hoare assertion $\{P\}\,\alpha\,\{Q\}$ has a simple interpretation in any category. Let $\alpha : X \to Y$ be a morphism and let the "tests" P, Q be monics $i : P \to X$, $j : Q \to Y$. Then the assertion holds precisely when there exists (necessarily unique) $\beta : P \to Q$ with $j\beta = \alpha i$. If $i_1 : P_1 \to X$ is another monic, write $i \leqslant i_1$ if there exists (necessarily unique) $k : P \to P_1$ with $i_1 k = i$. Such $\leqslant$ is a reflexive and transitive relation on the class of monics into X whose antisymmetry classes constitute the partially ordered class of *subobjects* of X. Hoare assertions are invariant under choice of subobject representative, so P, Q should be regarded as subobjects. If the pullback

$$\begin{tikzcd} {[\alpha]Q} \arrow[r] \arrow[d, "t"'] & Q \arrow[d, "j"] \\ X \arrow[r, "\alpha"'] & Y \end{tikzcd}$$

exists, t is monic and $\{P\}\,\alpha\,\{Q\} \Leftrightarrow P \leqslant [\alpha]Q$. The usual categories **Pfn** of sets and partial functions and **Rel** of sets and relations both have such pullbacks even though they fail to have pullbacks generally.

What axioms on a category will allow the tests to range over a Boolean algebra while the "predicate transformer" $[\alpha]Q$ will have the properties required by dynamic logic [8,10,22] and its precursor [4]? In a topos, the

* Tel.: +1 413 369 4246.
E-mail address: manes@mtdata.com

0304-3975/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2006.02.014

subobjects of X form a Brouwerian lattice whose Boolean center consists of the *summands* of X (the subobjects represented by a coproduct injection), an idea that meshed well with earlier ideas of [7] about program semantics using coproducts. In 1992 [13], the author introduced *Boolean categories* for this purpose (see Section 5 for the four axioms). A proof of Kozen [10] was adapted to prove the metatheorem [13, 11.15] that the universally valid formulas about predicate transformers in Boolean categories are precisely those which hold in the Boolean category **Rel**.

"Exceptions" and "deterministic morphisms" are discussed in the next two sections. **Pfn** has unique exceptions and all morphisms are deterministic. **Rel** has unique exceptions and its deterministic morphisms are precisely the partial functions. In spite of Kozen's theorem, Boolean categories can support multiple exceptions. For details we refer the reader to [13] with specific emphasis on its 22 page introduction.

We turn next to restriction categories. Partial functions have long been studied in computability theory and even earlier in algebraic geometry (not to mention freshman calculus!). The concept of "partial symmetry" enjoys diverse application and has received some careful attention (see [11]). The *restriction categories* introduced by Cockett and Lack [2] (whose four axioms are given in Section 4) provide an abstract model of partiality in categories. See their bibliography for many other category-theoretic models of partiality (references also appear at the beginning of Section 4), but none of these has the unusual property enjoyed by restriction categories that any full subcategory again is one. Their idea is to equationally axiomatize the "domain" of $f : X \to Y$ (classically, the set of x for which fx is defined) as an endomorphism $\bar{f} : X \to X$ (generalizing the guard function $\bar{f}x = x$ if fx is defined and $\bar{f}$ otherwise undefined). One observes in the culture of modern algebra that it is those structures with few equations and substantial consequence that have most likely permanence, and this author believes that "restriction" has such a future. See [18] for some development of the restriction axioms on a semigroup.

The most direct approach to model partiality in a category is to fix a class $\mathcal{M}$ of monics with appropriate properties and to form "partial morphisms" $X \to Y$ which are equivalence classes $[i, f]$ of $X \xleftarrow{i} A \xrightarrow{f} Y$ with $i \in \mathcal{M}$. Another representative $X \xleftarrow{j} B \xrightarrow{g} Y$ is equivalent when there exists an isomorphism $\psi : A \to B$ with $j\psi = i$ and $g\psi = f$. $[id, id]$ provides identity morphisms and the composition $X \xrightarrow{[i,f]} Y \xrightarrow{[j,g]} Z$ is defined using the pullback $[f]B$ (whose existence is part of the requirements on $\mathcal{M}$) as $[X \xleftarrow{i} A \longleftarrow [f]B \longrightarrow B \xrightarrow{g} Z]$. Given that restriction categories are closed under full subcategories, an arbitrary restriction category cannot be expected to have such pullbacks or to be closed under any limit construction for that matter. There is, however, a canonical class $\mathcal{M}$ for which an appropriate idempotent completion always has the needed pullbacks. [2, Theorem 3.4, Corollary 3.5] establishes the completeness theorem that every restriction category is a full subcategory of a category of partial maps as above.

The third construct concerns monads in program semantics. We expect the reader to know basic facts about monads (noting that none are needed for the first six sections of the paper). The survey article [17] is more than adequate; specific results referred there are, as a rule, proved elsewhere earlier by others. We say enough here to make this paper reasonably self-contained.

The most well-known definition of a *monad* $\mathbf{T}$ in a category $\mathcal{K}$ is $\mathbf{T} = (T, \eta, \mu)$ with $T : \mathcal{K} \to \mathcal{K}$ a functor, $\eta : id \to T$, $\mu : TT \to T$ natural transformations subject to the further equations $\mu(T\eta) = id = \mu(\eta T)$, $\mu(T\mu) = \mu(\mu T)$, seven equations in all. An alternate definition is $\mathbf{T} = (T, \eta, (-)^{\#})$ with $T : Obj(\mathcal{K}) \to Obj(\mathcal{K})$, η a family $\eta_X : X \to TX$ and $\alpha : X \to TY \mapsto \alpha^{\#} : TX \to TY$ subject to the three equations $\eta^{\#} = id$, $\alpha^{\#}\eta = \alpha$ and, for $\alpha : X \to TY$, $\beta : Y \to TZ$, $(\beta^{\#}\alpha)^{\#} = \beta^{\#}\alpha^{\#}$ in which T is never iterated. These definitions are equivalent. Given the alternate definition, T is a functor if $Tf = (X \xrightarrow{f} Y \xrightarrow{\eta_Y} TY)^{\#}$, and η is then natural. μ is defined by $\mu_X = (id_{TX})^{\#}$. In the other direction, define $\alpha^{\#} = TX \xrightarrow{T\alpha} TTY \xrightarrow{\mu_Y} TY$. Work on monads as used in Haskell favor the alternate definition (see [20,24]).

Algebras over a monad play a crucial role in understanding the relevance of monads in this paper. Our treatment is brief since this theory is not used after this introduction. For $\mathbf{T}$ a monad in $\mathcal{K}$, a $\mathbf{T}$-*algebra* is (X, ξ) with the *structure map* $\xi : TX \to X$ of the algebra satisfying $\xi\eta_X = id$ and $\xi\mu_X = \xi(T\xi)$. If (Y, θ) is another $\mathbf{T}$-algebra, a $\mathbf{T}$-homomorphism $f : (X, \xi) \to (Y, \theta)$ is a morphism $f : X \to Y$ with $\theta(Tf) = f\xi$. The resulting category of $\mathbf{T}$-algebras is written $\mathcal{K}^{\mathbf{T}}$. The monad axioms guarantee that (TX, μ_X) is a $\mathbf{T}$-algebra, and it is in fact the one freely generated by X in the sense that for $f : X \to TY$ any morphism there exists a unique $\mathbf{T}$-homomorphism $g : (TX, \mu_X) \to (TY, \mu_Y)$ with $g\eta_X = f$.

Such g is written $f^{\#}$ and is defined as $TX \xrightarrow{Tf} TY \xrightarrow{\theta} Y$; when (Y, θ) has form (Z, μ_Z) this agrees with the earlier $f^{\#}$. Thus the monad can be recovered from its free algebras. This is *Huber's Theorem* [17, 2.8].

Now let $\mathcal{K} = \mathcal{S}$, the category of sets and total functions. For $A \subset X$, $\omega \in TX$, A is a *support* of ω if $\omega \in TA$, that is, if $i : A \to X$ is inclusion, ω is in the image of $Ti : TA \to TX$. Write the set of supports of ω as $Supp(\omega) \subset 2^X$. Say that the monad $\mathbf{T}$ is *finitary* if for all X, any $\omega \in TX$ has a finite support. The concept of a category of algebras of a finitary monad is coextensive (via an isomorphism of categories over $\mathcal{S}$) with a variety of universal algebras presented by finitary operations and equations relating them [17, Theorem 3.44]. Moreover, if $\mathcal{S}^{\mathbf{T}}$ is presented by finitary operations and equations and $\mathcal{S}^{\mathbf{S}}$ is obtained by adding further equations in the same operations, $\mathcal{S}^{\mathbf{S}}$ is an epi-reflective full subcategory of $\mathcal{S}^{\mathbf{T}}$, that is, each $\mathbf{T}$-algebra admits a surjective $\mathbf{T}$-homomorphic reflection to an $\mathbf{S}$-algebra with the usual universal property. The reflection of the free $\mathbf{T}$-algebra produces the free $\mathbf{S}$-algebra, and this reflection $\gamma_X : TX \to SX$ constitutes a natural transformation which is, in fact, a monad map as defined in Section 7. This is the monadic form of the Birkhoff Variety Theorem [17, Proposition 3.8, Theorem 3.9]. The following examples will be useful momentarily. Let $\mathbf{L} = (L, \eta, (-)^{\#})$ be the list monad, $LX = X^*$, $\eta_X x = [x]$, $\alpha^{\#}[x_1, \ldots, x_n] = \alpha(x_1) + \cdots + \alpha(x_n)$ (where $+$ is list concatenation). Notice that μ_X flattens a list of lists to a list. As X^* is the free monoid generated by X, $\mathcal{S}^{\mathbf{L}}$ is the category of monoids and monoid homomorphisms. The usual presentation has a nullary operation e and a binary operation xy satisfying the equations $x(yz) = (xy)z$, $xe = x = ex$. Imposing the further equations $xy = yx$, $xx = x$ constructs the variety $\mathcal{S}^{\mathbf{S}}$ of semilattices with bottom. The free such semilattice is P_0X, the set of finite subsets of X. The surjective reflection map $\gamma_X : X^* \to P_0X$ maps $[x_1, \ldots, x_n]$ to $\{x_1, \ldots, x_n\}$.

The paper [14] discusses "collection monads" and their implementation issues (not necessarily restricted to functional programming languages). A monad $\mathbf{T}$ in $\mathcal{S}$ is a *collection monad* if some X exists such that TX has at least two elements and if the following two axioms hold: *members exist* (each $\omega \in TX$ has a finite minimum support called the set $mem(\omega)$ of *members* of ω) and *members are collected* (for $\alpha : X \to TY$, $\omega \in TX$, $mem(\alpha^{\#}\omega) = \bigcup_x mem(\alpha x)$). An *implementation* of a collection monad involves a tree-valued natural transformation (the actual definition [14, 6.1] is more complicated). A collection monad is *ordered* if in a suitably precise sense (to be given shortly below) its members can be canonically listed. It is a theorem that every implementable monad is ordered. This accords with the intuition that to implement a set, its elements must be listed in some order, so that the finite subsets monad is not (directly) implementable. What is somewhat surprising is that basic properties of collections can be characterized in terms of finite limits. For example, [14, Theorem 6.10] asserts that a collection monad is ordered if and only if T preserves equalizers of pairs of monics. As was the case for Boolean categories and restriction categories, certain pullbacks are important for collection monads as well, and we now turn in this direction.

The following terminology will find frequent use.

Definition 1.1. An arbitrary functor is *taut* if given a pullback

$$\begin{tikzcd} P \arrow[r] \arrow[d] & Q \arrow[d, "j"] \\ X \arrow[r] & Y \end{tikzcd}$$

with j monic, the image under the functor is again a pullback. A natural transformation is *taut* if each monic-induced naturality square is a pullback. A monad (T, η, μ) is *taut* if T, η, μ are.

The promised formal definition that a collection monad be ordered depends only on the functor T and it is this: there exists a taut list-valued natural transformation $T \to L$.

By [14, Theorem 4.14], a finitary monad is a collection monad if and only if the corresponding variety of algebras can be described with finitary operations and *balanced* equations, that is, with equations for which the same set of variables occurs in both terms. The fact that the three monoid equations are balanced (here, e has no variables so $xe = x$ is indeed balanced), is a proof that the list monad is a collection monad. Adding the balanced equation $xy = yx$ (so that repetition counts, but not order) gives bags, so that the free commutative monoid generated by X is the set of all bags on X. Adding yet the further balanced equation $xx = x$ then gives finite sets, as discussed earlier. So bags and sets are collection monads as well. How does this related to tautness? Let **T**-algebras be presented with finitary operations and no equations and let **S**-algebras be presented with the same operations and set E of equations, resulting in the pointwise-surjective natural transformation $\gamma : T \to S$. By [14, Theorem 4.14], E is balanced if and only if γ is taut and, in that case, T is taut as well.

The paper [16] makes further study of tautness which leads to a new characterization of collection monads as we now describe. If $2 = \{True, False\}$ and $\wedge : 2 \times 2 \to 2$ is Boolean "and" then $(2, \wedge, True)$ is a monoid. Let X be a set, $A \subset X$ with inclusion $i : A \to X$ and characteristic function $\chi_A : X \to 2$. Then, recalling that the algebras of the list monad **L** are monoids, $(\chi_A)^\# : X^* \to (2, \wedge, True)$ is the $fold \wedge$ operator, that is,

$$(\chi_A)^\#[x_1, \ldots, x_n] = (x_1 \in A) \wedge \cdots \wedge (x_n \in A) = (fold\wedge)[x_1, \ldots, x_n].$$

Observe that

$$\begin{tikzcd} A^* \arrow[r] \arrow[d, "i^*"'] & 1 \arrow[d, "True"] \\ X^* \arrow[r, "(\chi_A)^\#"'] & (2, \wedge, True) \end{tikzcd}$$

is a pullback in $\mathcal{S}$. For a general monad **T** in $\mathcal{S}$, a *support classifier* for **T** is a **T**-algebra of form $(2, \psi)$ such that for all inclusions $i : A \to X$ the square

$$\begin{tikzcd} TA \arrow[r] \arrow[d, "Ti"'] & 1 \arrow[d, "True"] \\ TX \arrow[r, "(\chi_A)^\#"'] & (2, \psi) \end{tikzcd}$$

is a pullback. This says that $A \in Supp(\omega) \Leftrightarrow (\chi_A)^\#(\omega) = True$, hence the term. Monad maps will be defined in Section 7. The filter monad **F** will be defined in Section 8. [16, Section 4] establishes that if a support classifier exists then it is unique and that the following theorem holds:

Theorem 1.2. *For a monad* $\mathbf{T} = (T, \eta, \mu)$ *in* $\mathcal{S}$, *the following conditions are equivalent*:
(1) **T** *has a support classifier.*
(2) *For* $\omega \in TX$, $Supp(\omega)$ *is a filter and* $Supp : \mathbf{T} \to \mathbf{F}$ *is a taut monad map.*
(3) *There exists a taut monad map* $\mathbf{T} \to \mathbf{F}$.
(4) **T** *is a taut monad.*

It is well known that **F**-algebras are continuous lattices. The two-element continuous lattice with $False < True$ provides a support classifier—one must exist since $id : \mathbf{F} \to \mathbf{F}$ is a taut monad map. For $\mathcal{F} \in FX$, $Supp(\mathcal{F}) = \mathcal{F}$ which shows that arbitrary filters arise as supports.

By now, the patient reader must be wondering what this all has to do with collection monads. We have

Theorem 1.3 (*Manes [16, Proposition 3.11]*). *Let* **T** *be a finitary monad in* $\mathcal{S}$. *Then* **T** *is a collection monad if and only if the conditions in Theorem* 1.2 *hold.*

This concludes the background. See the papers cited for examples. Now, what is this paper hoping to accomplish? There are two different goals. We begin with the first.

Although **Rel** is the "general" Boolean category, **Pfn** is an important example. The first project is to understand the overlap between Boolean categories and restriction categories. If $\mathcal{B}$ is a Boolean category, we seek a "natural" restriction operator for which "$\bar{f}x = x$ if fx is defined and $\bar{f}$ undefined otherwise". Now, in $\mathcal{B}$, each $f : X \to Y$ has a *kernel-domain decomposition* $K \xrightarrow{i} X \xleftarrow{j} D$ which is a coproduct; to avoid the need for further terminology at this point, it suffices to say that $K = [f]0$ whereas D is the largest summand $e : E \to X$ such that $fe = e$. The theory in [13] is needed to show that this exists and is a coproduct, but uniqueness is clear. Thus if

$$\begin{tikzcd} K = [f]0 \arrow[r, "i"] \arrow[d, "\alpha_f"'] & X \arrow[d, "f"] \\ 0 \arrow[r] & Y \end{tikzcd}$$

is a pullback, use the coproduct property to define $\bar{f}$ by $\bar{f}j = j$, $\bar{f}i = K \xrightarrow{\alpha_f} 0 \to X$. Unfortunately this will not work! The restriction category axioms require for $f : X \to Y$, $g : X \to Z$ that $\bar{f}\bar{g} = \bar{g}\bar{f}$. This fails in the following example whose proof we leave to the reader as an exercise in applying the results of this paper.

Example 1.4. Let $E = \{a, b\}$ be a two-element set of "exceptions". Let $\mathcal{B}$ be the category whose objects are sets and whose morphisms $X \to Y$ are functions $X \to Y + E$ where $+$ is coproduct in $\mathcal{S}$, i.e. disjoint union. Given $\beta : Y \to Z + A$ define $\beta^{\#} : Y + E \to Z + E$ by mapping y to βy and $e \in E$ to e. Such $\mathcal{B}$ is a category with the identity $X \to X + E$ being the first coproduct injection and with composition $\beta^{\#}\alpha$.

Then $\mathcal{B}$ is a Boolean category. Let $f, g : \{x_0\} \to \emptyset + E$ be defined by $f(x_0) = a$, $g(x_0) = b$. Then $\bar{g}\bar{f} = \bar{f}$, $\bar{f}\bar{g} = \bar{g}$ whereas $\bar{f} \neq \bar{g}$, so the axiom fails.

The problem in Example 1.4 is that the Boolean category there allows for multiple exceptions (see the next section for a formal definition of exception in a category). This possibility was intended from the beginning, but is incompatible with restriction. To remedy this, we look only at Boolean categories which have one exception, that is, whose initial object is in fact a zero object (see Proposition 2.3 and Observation 2.4). In that case, $\bar{f}$ is defined as above noting that α_f is a zero map. Having settled this point, we can announce the first of our main results, Theorem 6.4: a Boolean category with a zero object is such that the natural restriction satisfies the restriction category axioms if and only if all of its morphisms are deterministic (as defined in Section 4).

Boolean categories are intended to allow nondeterminism, whereas categories of partial maps do not. In the end, then, the result here is that the minimum necessary conditions turn out to be sufficient. The resulting categories are called *Boolean restriction categories*.

Our work leaves unanswered the interesting question as to what properties a given restriction category must have to be a Boolean restriction category. A satisfactory answer has been obtained in joint work with Robin Cockett which shall appear elsewhere.

We turn now toward the second goal of the paper. Associated with any monad $\mathbf{T}$ in $\mathcal{K}$ is its *Kleisli category* $\mathcal{K}_{\mathbf{T}}$ which is a new category with the same objects as $\mathcal{K}$ as discussed in Section 7. **Rel** and **Pfn** arise as Kleisli categories. In all nontrivial cases, $\mathcal{K}$ is a (nonfull) subcategory of $\mathcal{K}_{\mathbf{T}}$. If $\mathcal{K}$ is itself Boolean, we say $\mathbf{T}$ is a *Boolean monad* (Definition 9.1) if $\mathcal{K}_{\mathbf{T}}$ is a Boolean category containing $\mathcal{B}$ as a Boolean subcategory in a "standard" way (which very roughly asserts that if monad T is added on the fly, as in Haskell, the meaning of tests will not change).

The question, then, is this: for monads in $\mathcal{S}$ (which is a Boolean category, as is any topos) what is the relation between taut monads and Boolean monads? The surprising answer is that they are exactly the same! So if $\mathbf{T}$ is a finitary monad, what is needed to provide a theory of Hoare assertion following dynamic logic on $\mathcal{K}_{\mathbf{T}}$ is precisely that T be a collection monad.

2. Exceptions

Let $\mathcal{K}$ be any category.

Definition 2.1. For X an object, an X-*sink* is a family $\perp_Y : X \to Y$ indexed by the objects of $\mathcal{K}$ for which for all $z : Y \to Z$, $z \perp_Y = \perp_Z$. Say that $f : X \to Y$ is an *exception* if $f = \perp_Y$ for some X-sink $\perp$.

Lemma 2.2. *Given* $W \xrightarrow{f} X \xrightarrow{g} Y \xrightarrow{h} Z$ *with* g *an exception then* hgf *is an exception.*

Proof. There exists an X-sink $\perp$ with $g = \perp_Y$. Define

$$W \xrightarrow{\bullet_A} A = W \xrightarrow{f} X \xrightarrow{\perp_A} A.$$

Then, for $b : A \to B$, $b\bullet_A = (b \perp_A) f = \perp_B f = \bullet_B$, so $\bullet$ is a W-sink. Setting $b = h$, $hgf = h \perp_Y f = \bullet_Z$, so hgf is an exception. $\square$

Proposition 2.3. *For $\mathcal{K}$, equivalent are*
(1) *For each two objects X, Y there exists a unique exception $X \to Y$.*
(2) *$\mathcal{K}$ has zero morphisms.*
In that case, the zero morphisms are the unique exceptions.

Proof. (1) $\Rightarrow$ (2) Denote the unique exception by $0_{XY} : X \to Y$. If $f : W \to X$, $h : Y \to Z$ then $h0_{XY} f = 0_{XZ}$ by Lemma 2.2. (2) $\Rightarrow$ (1). For $\perp$ an X-sink, $\perp_Y = 0_{YY}$, so every exception is a zero. $\square$

Remark 2.4. If $\mathcal{K}$ has an initial object 0 then a morphism is an exception if and only if it factors through 0.

Proof. For any X-sink $\perp$,

$$X \xrightarrow{\perp_Y} Y = X \xrightarrow{\perp_0} 0 \longrightarrow Y$$

so every exception factors through 0. Conversely, for $g : X \to 0$,

$$\perp_Y = X \xrightarrow{g} 0 \longrightarrow Y$$

is an X-sink. $\square$

We are particularly interested in Boolean categories. These always have an initial object 0 and all $0 \to X$ are monic. Hence, in Boolean categories, exceptions factor uniquely through 0.

3. Total and deterministic maps

A morphism $f : X \to Y$ is *total* if for all $x : W \to X$, whenever fx is an exception then necessarily x is an exception.

Proposition 3.1. *Let $X \xrightarrow{f} Y \xrightarrow{g} Z$. Then*
(1) *If f, g are total so too is gf.*
(2) *If gf is total so too is f.*

Proposition 3.2. *Every monic is total.*

Proof. Let $f : X \to Y$ be monic and suppose $x : W \to X$ is such that there exists W-sink $\perp$ with $fx = \perp_Y$. As also $f \perp_X = \perp_Y$, $x = \perp_X$ is an exception. $\square$

Definition 3.3. A morphism $f : X \to Y$ is *deterministic* if it is impossible to witness two-valued behavior; more precisely, for all coproducts $Q \to Y \leftarrow Q'$ there exists a commutative diagram

$$\begin{tikzcd}
P \arrow[r] \arrow[d] & X \arrow[d, "f"] & P' \arrow[l] \arrow[d] \\
Q \arrow[r] & Y & Q' \arrow[l]
\end{tikzcd}$$

with the top row a coproduct.

The heuristics behind the definition require that coproducts behave like disjoint unions. This is the case in Boolean categories as is discussed in considerable detail in [13].

Remark 3.4. Deterministic morphisms form a subcategory.

4. Restriction categories

There have been many papers on categories of partial maps (see [1,3,5,12,19,21,23] and the references cited there). The motivating example, of course, is the category **Pfn** of sets and partial functions. In this category, for $A \subset X$ the construct $if x \in A...$ has semantics $g_A : X \to X$ where $g_A x = x$ if $x \in A$ and is otherwise undefined. Cockett and Lack considered a category with an additional unary operator $f : X \to Y \mapsto \bar{f} : X \to X$ generalizing $\bar{f} = g_A$ if $A = \{x : fx \text{ is defined}\}$.

Definition 4.1 (*Cockett and Lack [2]*). A *restriction category* is a category with a unary operator $\bar{f}$ as above subject to the following four equational axioms:
(R.1) $f\bar{f} = f$ for all $f : X \to Y$.
(R.2) For $f : X \to Y, g : X \to Z$, $\bar{f}\bar{g} = \bar{g}\bar{f}$.
(R.3) $\overline{g\bar{f}} = \bar{g}\bar{f}$ for all $f : X \to Y, g : X \to Z$.
(R.4) $\bar{g}f = f\overline{gf}$ for all $f : X \to Y, g : Y \to Z$.

In a category of partial functions with multiple notions of being undefined, $\bar{g}\bar{f}x$, if not defined, would be the sink state associated to $\bar{f}$. Thus the second axiom enforces "at most one exception". A formal counterexample was already given in Example 1.4.

It is an easy exercise to see that the category of sets and relations with $\bar{f}$ defined as in **Pfn** satisfies only the first three axioms, whereas the class of relations f satisfying the fourth axiom for all g is precisely the partial functions. Thus the fourth axiom (which first appears in [5]) expresses the requirement that all morphisms be deterministic. These relationships between (R.4) and deterministic morphisms is established formally in Theorem 6.4.

In a restriction category, $\bar{f}$ is idempotent because $\bar{f}\,\bar{f} = \overline{f\bar{f}} = \bar{f}$. In a *split* restriction category, each $\bar{f}$ splits.

A functor between restriction categories is a *restriction* functor if it preserves $\bar{f}$. A subcategory of a restriction category is a *restriction* subcategory if its inclusion functor is a restriction functor.

5. Boolean categories

Boolean categories are intended as a general framework for program semantics with arbitrary levels of inderminacy and exceptions, spotlighting the use of coproducts to express Boolean conditionals. The axioms are as follows.

Definition 5.1 (*Manes [13, Definition 4.7]*). A category $\mathcal{B}$ is *Boolean* if the following four axioms hold:
(B.1) $\mathcal{B}$ has finite coproducts including an initial object 0.
(B.2) Given $f : X \to Y$ and a coproduct injection $j : Q \to Y$ there exists a pullback

$$\begin{tikzcd} {[f]Q} \arrow[r] \arrow[d, "k"'] & Q \arrow[d, "j"] \\ X \arrow[r, "f"'] & Y \end{tikzcd}$$

with k a coproduct injection.
(B.3) A coproduct injection pulls back coproducts to coproducts.
(B.4) If the identity morphisms $X \to X \leftarrow X$ form a coproduct then $X = 0$.

Evidently, any extensive category is Boolean. *Pfn*, while Boolean and possessing products and coproducts, is not extensive since it has a zero object.

For the remainder of this section $\mathcal{B}$ is a Boolean category.

It is immediate that every topos is a Boolean category. Moreover, it is provable [13, Proposition 5.15] that coproduct injections in $\mathcal{B}$ are equalizers, so are monic in particular. Thus the coproduct injections form a stable system of monics (as required by Cockett and Lack [2]) and there is the corresponding category of partial morphisms with the same objects as $\mathcal{B}$.

Subobject equivalence classes of coproduct injections into X are called *summands* of X and the poset of all such is denoted $Summ(X)$.

Theorem 5.2 (*Manes [13, Theorem 5.11]*). *$Summ(X)$ is a Boolean algebra with least element* 0, *greatest element X, intersection by pullbacks and coproduct complement* (*provably unique*) *as complement. For $P, Q \in Summ(X)$ with $P \cap Q = 0$, the poset inclusions $P \to P \cup Q \leftarrow Q$ form a coproduct.*

Proposition 5.3 (*Manes [13, Propositions 6.13, 6.5, 6.11]*). *For all $f : X \to Y$ in $\mathcal{B}$, there exist unique $K, D \in Summ(X)$* (*called a kernel-domain decomposition of f*) *such that f restricted to K is an exception and f restricted to D is total. K is the pullback $[f]0$ and has the universal property that given $x : W \to X$ with fx an exception, then x factors uniquely through K.*

Such K is called the *kernel $Ker(f)$* of f. Using K' for Boolean complement, $D = Ker(f)'$ is the *domain $Dom(f)$* of f and is characterized as the largest summand restricted to which f is total.

Notice that

$$f : X \to Y \text{ is total } \Leftrightarrow Ker(f) = 0 \Leftrightarrow Dom(f) = X.$$

A functor F between Boolean categories is *Boolean* if it preserves finite coproducts and the pullbacks $[f]Q$ and is further *strict* if $FX = 0 \Rightarrow X = 0$. A *wide Boolean subcategory* of $\mathcal{B}$ is a subcategory $\mathcal{C}$ of $\mathcal{B}$ containing all isomorphisms (and hence all objects) and whose inclusion is a (necessarily strict) Boolean functor.

Proposition 5.4 (*Manes [13, Propositions 6.17, 12.11]*). *The subcategories $\mathcal{B}_{\mathrm{tot}}$, $\mathcal{B}_{\mathrm{det}}$ of total*, *respectively deterministic*, *morphisms are wide Boolean subcategories of $\mathcal{B}$.*

While Boolean categories may have many exceptions, there is a canonical notion of the Boolean subcategory induced by each individual exception type as is now explained.

Definition 5.5. A *projection system* for $\mathcal{B}$ is a family $\rho_{XP} : X \to P$ as X ranges over all objects and $P \in Summ(X)$ such that for all X and $P, Q \in Summ(X)$, $\rho_{XX} = 1$ and the following square commutes:

$$\begin{tikzcd} P \arrow[r] \arrow[d, "\rho_{P(P\cap Q)}"'] & X \arrow[d, "\rho_{XQ}"] \\ P \cap Q \arrow[r] & Q \end{tikzcd}$$

If ρ is a projection system, denote by $\mathcal{B}_\rho$ the subcategory with the same objects as $\mathcal{B}$ and all morphisms $f : X \to Y$ such that

$$X \xrightarrow{f} Y \xrightarrow{\rho_{Y0}} 0 = \rho_{X0}$$

for all X.

It is an easy exercise to check that the family $\rho_{X0} : X \to 0$ determines the rest of the projection system. Specifically, ρ_{XP} is determined by ρ_{P0} and $\rho_{P'0}$.

Theorem 5.6 (*Manes [13, Theorem 8.11]*). *Each $\mathcal{B}_\rho$ is a maximal wide Boolean subcategory of $\mathcal{B}$ with zero maps and the passage $\rho \mapsto \mathcal{B}_\rho$ establishes a bijection between projection systems and maximal wide Boolean subcategories with zero maps. Moreover*, *any Boolean subcategory with zero maps is contained in some $\mathcal{B}_\rho$.*

6. Boolean restriction categories

Let $\mathcal{B}$ be a Boolean category, ρ a projection system on $\mathcal{B}$ with corresponding maximal wide zero-subcategory $\mathcal{B}_\rho$.

Definition 6.1. The *natural restriction* on $\mathcal{B}_\rho$ is the unary operator $f : X \to Y \mapsto \bar{f} : X \to X$ defined as follows. Let $i : Dom(f) \to X$, $i' : Ker(f) \to X$. By the coproduct property, $\bar{f}$ is defined by $\bar{f}i = i$, $\bar{f}i' = 0$.

We need a few observations for the proof of the next result. First of all, in a category with a zero object, $0 : X \to Y$ is deterministic. For in Definition 3.3 let the top row be $X \to 0 \leftarrow 0$ and fill in with zero maps. In $\mathcal{B}$, $Ker(f)$, $Dom(f)$ forms a kernel-domain decomposition for $\bar{f}$ so that $Dom(\bar{f}) = Dom(f)$. (In more detail, because of zero maps, $\bar{f}i = i$ if $i : D \to X$ is the domain of f whereas $\bar{f}i' = 0$. This proves both that $f, \bar{f}$ have the same domain and also that $\bar{f} = \bar{g}$ if f, g have the same domain.) $\bar{f}$ is deterministic by [13, 8.12, p. 167]. Finally, we recall from [13] the *weakest precondition* operator for $f : X \to Y$, $Q \in Summ(Y)$

$$wp(f, Q) = [f]Q \cap Dom(f).$$

If g is deterministic, [13, Proposition 8.6(2)] gives that $Dom(fg) = wp(g, Dom(f))$.

Proposition 6.2. *The natural restriction on $\mathcal{B}_\rho$ satisfies the first three axioms* (R.1, R.2, R.3) *for restriction categories. Moreover, each $\bar{f}$ splits.*

Proof. We use the notation of Definition 6.1. As $f\bar{f}i = fi$ and $f\bar{f}i' = f0 = 0 = fi'$, it follows from the uniqueness of coproduct-induced maps that $f\bar{f} = f$ which is (R.1). Turning toward (R.2), it is clear that if i is the inclusion of $Dom(\bar{f}\bar{g})$ then $\bar{f}\bar{g}i = i$. But by Manes [13, Theorem 8.15], $Dom(\bar{f}\bar{g}) = Dom(\bar{f}) \cap Dom(\bar{g})$; (R.2) follows. For (R.3), it is equivalent to show $Dom(f\bar{g}) = Dom(f) \cap Dom(g)$. We have

$$\begin{aligned} Dom(f\bar{g}) &= wp(\bar{g}, Dom(f)) \\ &= Dom(\bar{g}) \cap [\bar{g}](Dom(f)) \\ &= Dom(g) \cap (Ker(\bar{g}) \cup Dom(f)) \; [13, 10.1(9)] \\ &= (Dom(g) \cap Ker(g)) \cup (Dom(g) \cap Dom(f)) \\ &= Dom(f) \cap Dom(g). \end{aligned}$$

Finally, by the uniqueness of coproduct-induced maps, $X \xrightarrow{\rho} Dom(f) \xrightarrow{i} X = \bar{f}$. As $i\rho i = i$ with i monic, $\rho i = 1$ so $\bar{f}$ splits. $\square$

Definition 6.3. A *Boolean restriction category* is a Boolean category with zero morphisms in which all morphisms are deterministic.

We now establish the first of our main results, that a Boolean restriction category is indeed a restriction category.

Theorem 6.4. *In the context of Proposition* 6.2, *axiom* (R.4) $f\overline{gf} = \bar{g}f$ *holds if and only if all morphisms are deterministic. Thus every Boolean restriction category is a split restriction category.*

Proof. First suppose that (R.4) holds. Let $g : Y \to Z$ be arbitrary. To show that g is deterministic, it suffices to show that $j : Dom(g) \to Y$ is the *totalizer* of g, that is, that gj is total whereas for all $x : X \to Y$ with gx total, x factors through j; this is by Manes [13, Proposition 12.10]. For such x, $\overline{gx} = 1$. Applying (R.4), $\bar{g}x = x\overline{gx} = x$. Define $\varphi = \rho x$ where $\bar{g} = Y \xrightarrow{\rho} Dom(g) \xrightarrow{j} Y$. Then $j\varphi = j\rho x = \bar{g}x = x$ as needed. Conversely, assume all morphisms are deterministic and show (R.4) for $X \xrightarrow{f} Y \xrightarrow{g} Z$. Let $i : Dom(f) \to X$, $t : Dom(gf) \to Dom(f)$, $j : Dom(g) \to Y$, $j' : Ker(g) \to Y$ be the summand inclusions. As $Dom(gf) = wp(f, Dom(g)) = [f](Dom(g))$ (as $Dom(gf) \subset Dom(f)$), there is a pullback square, the right hand square in the diagram immediately below. We use the coproduct $Ker(gf) \xrightarrow{k'} X \xleftarrow{k} Dom(gf)$ to show $f\overline{gf} = \bar{g}f$. The coordinates of the first map are given by $f\overline{gf}k = fk$, $f\overline{gf}k' = 0$. Now consider the diagram in which ψ is to be constructed.

$$\begin{tikzcd}
Ker(gf) \arrow[r, "k'"] \arrow[d, "\psi"'] & X \arrow[d, "f"] & Dom(gf) \arrow[l, "k"'] \arrow[d, "u"] \\
Ker(\bar{g}) \arrow[r, "j'"'] & Y & Dom(g) \arrow[l, "j"]
\end{tikzcd}$$

Then $\bar{g}fk = \bar{g}ju = ju = fk$. As $gfk' = 0$, fk' factors through $Ker(g) = Ker(\bar{g})$ inducing ψ by the pullback property. Thus $\bar{g}fk' = \bar{g}j'\psi = 0$. □

We note that in a Boolean restriction category the two notions of total for $f : X \to Y$ coincide:

$$\bar{f} = 1 \Leftrightarrow Dom(f) = X \Leftrightarrow f \text{ is total.}$$

Definition 6.5. Let $\mathcal{B}$ be a Boolean category. The *partial morphism category* $\mathcal{B}_{\text{part}}$ of $\mathcal{B}$ has the same objects as $\mathcal{B}$ and with a morphism from X to Y being an equivalence class $[P \xleftarrow{i} X \xrightarrow{f} Y]$ where i is a summand and f is total, and the equivalence relation is the obvious one, namely there exists an isomorphism out of P preserving the two maps. Composition is the usual definition

$$[Y \xleftarrow{j} Q \xrightarrow{g} Y] \circ [P \xleftarrow{i} X \xrightarrow{f} Y]$$

using the pullback $[f]Q$.

Notice [2, Proposition 3.1] that $\mathcal{B}_{\text{part}}$ is a split restriction category with

$$\overline{[P \xleftarrow{i} X \xrightarrow{f} Y]} = [X \xleftarrow{i} P \xrightarrow{i} X].$$

For a Boolean category, the passage $\mathcal{B} \to \mathcal{B}_{\text{part}}$ given by $f : X \to Y \mapsto [X \xleftarrow{i} Dom(f) \xrightarrow{fi} Y]$ is *not* functorial in general. But we have

Proposition 6.6. *For $\mathcal{B}$ a Boolean restriction category, the above construction $\mathcal{B} \to \mathcal{B}_{\text{part}}$ is functorial and is an isomorphism of restriction categories.*

Proof. This is an immediate corollary of [2, Theorem 3.4]. This is because, as was observed above at the end of the proof of Proposition 6.2, the class of monics that arise in splitting an $\bar{f}$ in $\mathcal{B}$ is precisely the class of summands. □

7. Kleisli categories

In this section, $\mathcal{K}$ is any category and $\mathbf{T} = (T, \eta, (-)^{\#})$ is a monad in $\mathcal{K}$.

Definition 7.1. The *Kleisli category* [9] $\mathcal{K}_{\mathbf{T}}$ of the monad $\mathbf{T}$ in $\mathcal{K}$ is the category with the same objects as $\mathcal{K}$, with morphisms $\mathcal{K}_{\mathbf{T}}(X, Y) = \mathcal{K}(X, TY)$ and with composition for $\alpha : X \to TY$, $\beta : Y \to TZ$ in $\mathcal{K}$ given by $\beta \circ \alpha = \beta^{\#}\alpha : X \to TZ$.

Unless otherwise noted, displayed arrows are in $\mathcal{K}$. Composition in $\mathcal{K}$ will be denoted by juxtaposition and Kleisli composition by $\circ$.

Under any of the equivalent conditions of Proposition 7.3, $\mathcal{K}_{\mathbf{T}}$ is equivalent to the full subcategory of $\mathcal{K}^{\mathbf{T}}$ of all free $\mathbf{T}$-algebras, via the isomorphism $\alpha \mapsto \alpha^{\#}$. For example, Eilenberg [6, p. 2] defines a relation as a union-preserving map (that is, as a complete semilattice homomorphism $\alpha^{\#}$) rather than as a function $\alpha : X \to 2^Y$.

The following constructions will be used without further comment below. There is a canonical functor $I : \mathcal{K} \to \mathcal{K}_{\mathbf{T}}$ defined by $IX = X$, $I(X \xrightarrow{f} Y) = X \xrightarrow{f} Y \xrightarrow{\eta_Y} TY$. We use the abbreviation $f^\bullet$ for If. A morphism of form $f^\bullet$ is said to be a *base morphism*. $T : \mathcal{K} \to \mathcal{K}$ is functorial if $Tf = (f^\bullet)^\#$. $\eta : 1 \to T$ is a natural transformation and, if $\mu_X = (1_{TX})^\#$, $\mu : TT \to T$ is natural as well. I has right adjoint given by $X \mapsto TX$, $\alpha \mapsto \alpha^\#$ so that I preserves all colimits which exist in $\mathcal{K}$.

Definition 7.2. If $\mathbf{S}$, $\mathbf{T}$ are monads in $\mathcal{K}$ (we use the same symbols η and $(-)^\#$ for both monads), a *monad map* $\lambda : \mathbf{S} \to \mathbf{T}$ is a family of morphisms $\lambda_X : SX \to TX$ satisfying $\lambda_X \eta_X = \eta_X$ and such that the following square commutes for arbitrary $\alpha : X \to TY$:

$$\begin{tikzcd} SX \arrow[r, "\lambda_X"] \arrow[d, "\alpha^\#"'] & TX \arrow[d, "(\lambda_Y \alpha)^\#"] \\ SY \arrow[r, "\lambda_Y"'] & TY \end{tikzcd}$$

Notice that if $\lambda : \mathbf{S} \to \mathbf{T}$ is a monad map, it induces a functor $\mathcal{K}_{\mathbf{S}} \to \mathcal{K}_{\mathbf{T}}$ which is the identity on objects and maps $\alpha : X \to SY$ to $\lambda_Y \alpha$.

Proposition 7.3 (*Manes [17, Proposition 2.25]*). *For a monad* $\mathbf{T}$ *in* $\mathcal{K}$, *the following three conditions are equivalent. If any, and hence all, hold we say* $\mathbf{T}$ *is* nontrivial.
(1) η *is pointwise monic.*
(2) $I : \mathcal{K} \to \mathcal{K}_{\mathbf{T}}$ *is faithful.*
(3) $T : \mathcal{K} \to \mathcal{K}$ *is faithful.*

Proposition 7.4. *The following hold*:
(1) *For* $f : X \to Y$, $\beta : Y \to TZ$, $\beta \circ f^\bullet = \beta f$.
(2) *For* $\alpha : X \to TY$, $g : Y \to Z$, $g^\bullet \circ \alpha = (Tg)\alpha$.
(3) [13, Lemma 16.10 (2)] *Consider the two squares shown below, the first in* $\mathcal{K}_{\mathbf{T}}$ *and the second in* $\mathcal{K}$. *Then one commutes if and only if the other does and one is a pullback if and only if the other is.*

$$\begin{tikzcd} W \arrow[r, "a"] \arrow[d, "b"'] & X \arrow[d, "c"] \\ Y \arrow[r, "d"'] & Z \end{tikzcd} \qquad \begin{tikzcd} TW \arrow[r, "a^\#"] \arrow[d, "b^\#"'] & TX \arrow[d, "c^\#"] \\ TY \arrow[r, "d^\#"'] & TZ \end{tikzcd}$$

Proposition 7.5. *For* $\alpha : X \to TY$, α *is an isomorphism in* $\mathcal{K}_{\mathbf{T}}$ *if and only if* $\alpha^\# : TX \to TY$ *is an isomorphism in* $\mathcal{K}$.

Proof. First suppose $\beta : Y \to TX$ is inverse to α in $\mathcal{K}_{\mathbf{T}}$. As $\beta \circ \alpha = \eta_X$, $1_{TX} = (\eta_X)^\# = (\beta^\# \alpha)^\# = \beta^\# \alpha^\# = 1_{TY}$. Conversely, let $\psi : TY \to TX$ be inverse to $\alpha^\#$ in $\mathcal{K}$. Consider the diagrams

$$\begin{tikzcd} TTX \arrow[r, "T\alpha^\#"] \arrow[d, "\mu_X"'] \arrow[dr, phantom, "\text{(A)}"] & TTY \arrow[r, "T\psi"] \arrow[d, "\mu_Y"] \arrow[dr, phantom, "\text{(B)}"] & TTX \arrow[d, "\mu_X"] \\ TX \arrow[r, "\alpha^\#"'] & TY \arrow[r, "\psi"'] & TX \end{tikzcd}$$

As $\alpha^\#$ is a $\mathbf{T}$-homomorphism [17, Proposition 2.17], (A) commutes for any α. As $\psi \alpha^\# = 1$ the perimeter commutes. But $T\alpha^\#$ is split epic, so (B) commutes which says that ψ is a $\mathbf{T}$-homomorphism and hence (by the proposition just cited) that $\psi = \beta^{\#}$ for $\beta = \psi\eta_Y$. We have

$$\eta_X = \eta_X(\eta_X)^{\#} = \eta_X\beta^{\#}\alpha^{\#} = \eta_X(\beta^{\#}\alpha)^{\#} = \beta^{\#}\alpha = \beta \circ \alpha. \qquad \square$$

Since I preserves coproducts, the poset $Summ(X)$ in $\mathcal{K}$ is a subposet of $Summ(X)$ in $\mathcal{K}_{\mathbf{T}}$. The latter can, in fact, be much larger. For example, let $\mathcal{K} = \mathcal{S}$, let TX be the real vector space with basis X with basis inclusion η_X and let $\alpha^{\#}$ be the unique linear extension of α. This is a monad. Now consider a two-element set 2. $T2$ is the plane. In $\mathcal{S}$, $Summ(2)$ is the four-element Boolean algebra. In $\mathcal{S}_{\mathbf{T}}$, each nonzero vector v in the plane generates a summand $v : 1 \to T2$ with two such being equivalent if and only if they lie on the same line through the origin. Hence in the Kleisli category, $Summ(2)$ is an uncountable flat poset with top and bottom. The usual x- and y-axes embed the four-element Boolean algebra.

Definition 7.6. For $\mathbf{T}$ a monad in $\mathcal{K}$, say that *summands are standard* in $\mathcal{K}_{\mathbf{T}}$ if $Summ(X)$ is the same poset in both categories. More precisely, for every coproduct injection $\alpha : P \to TX$ in $\mathcal{K}_{\mathbf{T}}$ there exists a coproduct injection $i : A \to X$ in $\mathcal{K}$ and $\gamma : P \to TA$ with γ an isomorphism in $\mathcal{K}_{\mathbf{T}}$ and with $i^{\bullet} \circ \gamma = \alpha$, that is, the following triangle commutes:

$$\begin{tikzcd}
P \arrow[rr, "\alpha"] \arrow[dr, "\gamma"'] & & TX \\
& TA \arrow[ur, "Ti"'] &
\end{tikzcd}$$

Proposition 7.7. *Let $\mathcal{K}$ have binary coproducts and let $\mathbf{T}$ be a nontrivial monad in $\mathcal{K}$ such that all isomorphisms in $\mathcal{K}_{\mathbf{T}}$ are base. Then summands are standard.*

Proof. Let $\alpha : P \to TX$ be a summand in $\mathcal{K}_{\mathbf{T}}$ so that there exists a coproduct $P \xrightarrow{\alpha} TX \xleftarrow{\alpha'} P'$ in $\mathcal{K}_{\mathbf{T}}$. As I preserves the coproduct $P \xrightarrow{i} P + P' \xleftarrow{i'} P'$ in $\mathcal{K}$ and as isomorphisms are base, there exists $g : P + P' \to X$ in $\mathcal{K}$ with $g^{\bullet}$ an isomorphism in $\mathcal{K}_{\mathbf{T}}$ such that

$$\begin{tikzcd}
P \arrow[rr, "i^{\bullet}"] \arrow[dr, "\alpha"'] & & T(P + P') \arrow[dl, "Tg"] \\
& TX &
\end{tikzcd}$$

Thus

$$\alpha = (Tg)i^{\bullet} = (Tg)\eta_{P+P'}i = \eta_X gi = (gi)^{\bullet}$$

shows that α is base. As η is monic, g is an isomorphism in $\mathcal{K}$. Thus $gi : P \to X$ is a summand in $\mathcal{K}$. Since $\eta_X(gi) = T(gi)\eta_P$, we can choose $\gamma = \eta_P$, the identity map of P in $\mathcal{K}_{\mathbf{T}}$ as the isomorphism in Definition 7.6. $\square$

The converse of Proposition 7.7 does not hold as the following example shows.

Example 7.8. Let G be a group with at least two elements with unit e. Define a monad $\mathbf{T}$ in $\mathcal{S}$ as follows. Let $TX = G \times X$ be the underlying set of the free left G-set generated by X. The left action on TX is $g, (h, x) \mapsto (gh, x)$ and the unique equivariant extension of $\alpha : X \to G \times Y$ through $\eta_X : X \to G \times X$, $x \mapsto (e, x)$ is given by $\alpha^{\#} : G \times X \to G \times Y$, $(g, x) \mapsto (gh, y)$ if $\alpha x = (h, y)$. We shall show that while not every isomorphism in $\mathcal{K}_{\mathbf{T}}$ is base, summands are standard nonetheless.

If $g \in G$, $g \neq e$, the left translation $\psi : G \to G$, $\psi h = hg$ is equivariant, hence is an isomorphism. But the only base morphism $G \to G$ is id because $G = T1$. This shows that not every isomorphism in $\mathcal{S}_{\mathbf{T}}$ is base.

Now let $\alpha : X \to G \times Y$ and let A be the image of the equivariant map $\alpha^{\#} : G \times X \to G \times Y$. If $(g, y) \in A$ and $h \in G$ then $(h, y) = hg^{-1}(g, y) \in A$. This shows that A has form $G \times Q$ for some $Q \subset Y$. Thus $\alpha^{\#}$ has unique factorization

$$\begin{tikzcd} G \times X \arrow[rr, "\alpha^{\#}"] \arrow[dr, "\psi"'] & & G \times Y \arrow[dl, "1 \times i"] \\ & G \times Q & \end{tikzcd}$$

where $i : Q \to Y$ is inclusion. If $\beta : X \to G \times Q$ is $\psi\eta_X$, $\psi = \beta^{\#}$ because ψ is equivariant. As a result, if $\alpha^{\#}$ were monic then $\beta^{\#}$ is bijective and so β is an isomorphism in $\mathcal{S}_{\mathbf{T}}$ by Proposition 7.4. As

$$\begin{tikzcd} X \arrow[rr, "\alpha"] \arrow[dr, "\beta"'] & & Y \arrow[dl, "i^{\bullet}"] \\ & Q & \end{tikzcd}$$

commutes in $\mathcal{S}_{\mathbf{T}}$ we have shown: for $\alpha : X \to G \times Y$ with $\alpha^{\#}$ monic, α is a standard summand. To this end, let $P \xrightarrow{\alpha} G \times X \xleftarrow{\alpha'} P'$ be a coproduct in $\mathcal{S}_{\mathbf{T}}$. We must show that $\alpha^{\#}$ is monic. There exists an isomorphism $\gamma : X \to P + P'$ in $\mathcal{S}_{\mathbf{T}}$ with $\gamma \circ \alpha = (in_P)^{\bullet}$. Equivalently,

$$\begin{tikzcd} G \times P \arrow[rr, "\alpha^{\#}"] \arrow[dr, "1 \times in_P"'] & & G \times X \arrow[dl, "\gamma^{\#}"] \\ & G \times (P + P') & \end{tikzcd}$$

But as $1 \times in_P$ is monic, so is $\alpha^{\#}$.

Definition 7.9. Taut functors, natural transformations and monads were defined in Definition 1.1. A *taut monad map* is a monad map which is a taut natural transformation.

There are only two trivial monads in $\mathcal{S}$. For one, $TX = 1$ for all X and for the other $TX = 1$ for all nonempty X whereas $T\emptyset = \emptyset$. Neither is a taut monad since in both cases η is not taut.

It is shown in [16, Proposition 2.2] that $\mathbf{T}$ is taut if and only if η is taut and, in the diagram below, if (A) is a pullback with i, j monic then (B) is also a pullback.

$$\begin{tikzcd} P \arrow[r, "\alpha"] \arrow[d, "i"'] \arrow[dr, phantom, "\text{(A)}"] & TX \arrow[d, "Tj"] \arrow[dr, phantom, "\text{(B)}"] & TP \arrow[l, "\alpha^{\#}"'] \arrow[d, "Ti"] \\ X \arrow[r, "\beta"'] & TY & TX \arrow[l, "\beta^{\#}"] \end{tikzcd}$$

Proposition 7.10. *If there exists a taut monad map* $\mathbf{S} \to \mathbf{T}$ *and if* $\mathbf{T}$ *is taut then* $\mathbf{S}$ *is taut.*

Proof. That η is taut is routine. The second axiom is proved in [16, Lemma 2.5]. $\square$

For the balance of this section, we assume that $\mathcal{K}$ has binary coproducts and that coproduct injections are monic.

Definition 7.11. α in $\mathcal{K}_{\mathbf{T}}$ is *standard-deterministic* if for every coproduct $Q \xrightarrow{j} Y \xleftarrow{j'} Q'$ in $\mathcal{K}$ there exists a coproduct $P \xrightarrow{i} Y \xleftarrow{i'} P'$ in $\mathcal{K}$ and morphisms β, β' such that the following diagram commutes in $\mathcal{K}$:

$$\begin{tikzcd} P \arrow[r, "i"] \arrow[d, "\beta"'] & X \arrow[d, "\alpha"] & P \arrow[l, "i'"'] \arrow[d, "\beta'"] \\ TQ \arrow[r, "Tj"'] & TY & TQ' \arrow[l, "Tj'"] \end{tikzcd}$$

Note that standard-deterministic maps are deterministic. For if $Q \xrightarrow{t} TY \xleftarrow{t'} Q'$ is any coproduct, there is an isomorphism $\psi : Q + Q' \to TY$ with $\psi^{\#}Ti = t^{\#}$, $\psi^{\#}Ti' = (t')^{\#}$. So, applying the definition of standard-deterministic to $X \xrightarrow{\alpha} TY \xrightarrow{(\psi^{\#})^{-1}} (Q + Q')$ produces P, P', g, g' with $t^{\#}g = \alpha i$, $(t')^{\#}g' = fi'$. Clearly standard-deterministic maps coincide with deterministic maps when summands are standard.

Lemma 7.12. *Let $\lambda : \mathbf{S} \to \mathbf{T}$ be a monad map with induced functor $H : \mathcal{K}_{\mathbf{S}} \to \mathcal{K}_{\mathbf{T}}$. Then if $\alpha : X \to SY$ is standard-deterministic in $\mathcal{K}_{\mathbf{S}}$, $H\alpha = \lambda_Y\, \alpha$ is standard-deterministic in $\mathcal{K}_{\mathbf{T}}$ and the converse holds if λ is taut.*

Proof. This is clear from the following diagram, noting that (A), (A$'$) are pullbacks when λ is taut.

$$\begin{tikzcd} P \arrow[r, "i"] \arrow[d] & X \arrow[d, "\alpha"] & P' \arrow[l, "i'"'] \arrow[d] \\ SQ \arrow[r, "Si"] \arrow[d, "\lambda_Q"'] \arrow[dr, phantom, "\text{(A)}"] & SY \arrow[d, "\lambda_Y"] \arrow[dr, phantom, "\text{(A}'\text{)}"] & SQ' \arrow[l, "Si'"'] \arrow[d, "\lambda_{Q'}"] \\ TQ \arrow[r, "Tj"'] & TY & TQ' \arrow[l, "Tj'"] \end{tikzcd}$$

$\square$

Lemma 7.13. *Let* $\mathbf{T}$ *be a monad such that η is taut. Then given $\alpha : X \to TY$, $\beta : Y \to TZ$ with β and $\beta \circ \alpha$ base, α is also base.*

Proof. Let $\beta = t^{\bullet}$, $\beta \circ \alpha = h^{\bullet}$. As $\beta^{\#} = Tt$ we have a pullback square

$$\begin{tikzcd} Y \arrow[r, "\eta_Y"] \arrow[d, "t"'] & TY \arrow[d, "\beta^{\#}"] \\ Z \arrow[r, "\eta_Z"'] & TZ \end{tikzcd}$$

By the pullback property there exists unique $f : X \to Y$ with $tf = h$ and $\eta_Y f = \alpha$. The second equation asserts that $\alpha = f^{\bullet}$ as required. $\square$

8. Taut monads of sets

In this section, all monads are in the category $\mathcal{S}$ of sets and total functions.

Example 8.1. Let $\mathbf{F}$ be the *filter monad* given by $FX = \{\mathcal{F} \subset 2^X : \mathcal{F} \text{ is a filter}\}$, $\eta_X(x) = prin(x) = \{A \subset X : x \in A\}$, $\alpha^\#(\mathcal{F}) = \{B \subset Y : \{x \in X : B \in \alpha(x)\} \in \mathcal{F}\}$.

Note that for $f : X \to Y$, $(Ff)\mathcal{F} = \{B \subset Y : f^{-1}(B) \in \mathcal{F}\}$. $\mathbf{F}$ is a taut monad and every one of its submonads has taut inclusion and so is taut as a monad [16, Example 2.4, Proposition 3.12].

Lemma 8.2. *Let* $\mathbf{T}$ *be any submonad of the filter monad. The following hold*:
(1) *If the proper filter* $2^X \in TX$ *then for all* $\alpha : X \to TY$, $\alpha^\#(2^X) = 2^Y$.
(2) *Let* $\mathcal{F} \in TX$ *and let* $\alpha : X \to TY$ *with* $\alpha^\#$ *injective. Then if* $\alpha^\#(\mathcal{F})$ *has form* $prin(y)$, $\mathcal{F}$ *has form* $prin(x)$.
(3) *Isomorphisms in* $\mathcal{S}_\mathbf{F}$ *are base.*

Proof. (1) $B \in \alpha^\#(2^X) \Leftrightarrow \{x \in X : B \in \alpha(x)\} \in 2^X$ and this is true for all B.

(2) For $B \subset Y$, $y \in B \Leftrightarrow \{x \in X : B \in \alpha(x)\} \in \mathcal{F}$. Setting $B = \{y\}$, $F = \{x \in X : \{y\} \in \alpha(x)\} \in \mathcal{F}$. If $F = \emptyset$ then $\mathcal{F} = 2^X \Rightarrow \alpha^\#(\mathcal{F}) = 2^Y$ by (1), but this is impossible as $\emptyset \notin prin(y)$. Let $x \in F$. As $\{y\} \in \alpha(x)$, $\alpha(x) = prin(y)$ or $\alpha(x) = 2^Y$. If $\alpha(x) = 2^Y$ then $\alpha^\#(prin(x)) = \alpha(x) = 2^Y = \alpha^\#(2^X)$ which violates the injectivity of $\alpha^\#$. Thus $\alpha(x) = prin(y)$. As $\mathbf{T}$ is nontrivial, α is injective. If $w \in F$, $\alpha(x) = prin(y) = \alpha(w)$ so $x = w$ and $F = \{x\} \in \mathcal{F}$. As $\mathcal{F} \neq 2^X$, $\mathcal{F} = prin(x)$.

(3) Let $\psi : X \to TY$ be an isomorphism in $\mathcal{S}_\mathbf{T}$. If $\varphi = \psi^{-1}$, $\varphi^\#$, $\psi^\#$ are mutually inverse bijections. Let $x \in X$. Define $\mathcal{F} = \psi^\#(prin(x))$ so that $\varphi^\#(\mathcal{F}) = prin(x)$. By (2), there exists $fx \in Y$ with $\mathcal{F} = prin(fx)$, defining a function $f : X \to Y$. By construction, $\psi^\#\eta_X = \eta_X f$. By the naturality of η, $(Tf)\eta_X = \eta_X f$. Applying [17, Proposition 2.19], $\psi^\# = Tf$ so $\psi = f^\bullet$ is a base morphism. $\square$

Lemma 8.3. *Let* $\mathbf{T}$ *be taut with corresponding taut monad map Supp* : $\mathbf{T} \to \mathbf{F}$ *as guaranteed by Theorem* 1.2. *The following hold*:
(1) $\alpha : X \to TY$ *is standard-deterministic if and only if for all* $x \in X$, *Supp*$(\alpha(x))$ *is either an ultrafilter or* 2^Y.
(2) *Every isomorphism in* $\mathcal{S}_\mathbf{T}$ *is standard-deterministic.*

Proof. (1) By Lemma 7.12, α is standard-deterministic if and only if $X \xrightarrow{\alpha} TY \xrightarrow{Supp_Y} FY$ is deterministic in $\mathcal{S}_\mathbf{F}$. To this end, let $Q \subset Y$ and consider the diagram

$$\begin{tikzcd}
P \arrow[r, "i"] \arrow[d] & X \arrow[d, "Supp_Y \alpha"] & P' \arrow[l, "i'"'] \arrow[d] \\
FQ \arrow[r, "Fj"'] & FY & FQ' \arrow[l, "Fj'"]
\end{tikzcd}$$

in which the top row is to be constructed. As $Fj = (j^\bullet)^\#$, it is easy to see that FQ consists of those filters on Y containing Q. P and its complement P' as needed will exist if and only if

$$(\forall x \in X) Q \in Supp(\alpha(x)) \text{ or } Q' \in Supp(\alpha(x)).$$

As Q is arbitrary this happens if and only if *Supp*$(\alpha(x))$ is either an ultrafilter or 2^Y.

(2) If $\alpha : X \to TY$ is an isomorphism in $\mathcal{S}_\mathbf{T}$ then $Supp_Y \alpha$ is an isomorphism in $\mathcal{S}_\mathbf{F}$ and hence is base by Lemma 8.2(3). Thus $Supp_Y \alpha(x)$ is a principal ultrafilter for all x. By (1), the proof is complete. $\square$

We can now establish the main result of this section.

Theorem 8.4. *Let* $\mathbf{T}$ *be a taut monad. Then summands are standard in* $\mathcal{S}_\mathbf{T}$.

Proof. Let $P \xrightarrow{\alpha} TX \xleftarrow{\alpha'} P'$ be an arbitrary coproduct decomposition of X in $\mathcal{S}_{\mathbf{T}}$. If $P \xrightarrow{j} P + P' \xleftarrow{j'} P'$ is a coproduct in $\mathcal{S}$, there exists a unique $\psi : X \to T(P + P')$ with $\psi^{\#}\alpha = j^{\bullet}$, $\psi^{\#}\alpha' = (j')^{\bullet}$ and such ψ is an isomorphism in $\mathcal{S}_{\mathbf{T}}$. By Lemma 8.3(2), ψ is standard-deterministic. Thus there exists a commutative diagram

$$\begin{tikzcd} R \arrow[r, "i"] \arrow[d, "\beta"'] \arrow[dr, phantom, "(A)"] & X \arrow[d, "\psi"] & R' \arrow[l, "i'"'] \arrow[d, "\beta'"] \\ TP \arrow[r, "Tj"'] & T(P+P') & TP' \arrow[l, "Tj'"] \end{tikzcd}$$

where the top row is a coproduct. Consider the squares

$$\begin{tikzcd} R \arrow[r, "i"] \arrow[d, "\beta"'] \arrow[dr, phantom, "(B)"] & X \arrow[d, "\eta_X"] \\ TP \arrow[r, "\alpha^{\#}"'] & TX \end{tikzcd} \qquad \begin{tikzcd} TR \arrow[r, "Ti"] \arrow[d, "\beta^{\#}"'] \arrow[dr, phantom, "(A)^{\#}"] & TX \arrow[d, "\psi^{\#}"] \\ TP \arrow[r, "Tj"'] & T(P+P') \end{tikzcd}$$

Then (B) commutes because

$$\psi^{\#}\alpha^{\#}\beta = (\psi^{\#}\alpha)^{\#}\beta = (j^{\bullet})^{\#}\beta = (Tj)\beta = \psi i = \psi^{\#}\eta_X i$$

and $\psi^{\#}$ is bijective, monic in particular. (A)$^{\#}$ commutes because (A) does [17, Proposition 2.19]. T preserves monics for any monad in $\mathcal{S}$ [14, Lemma 2.8], so it follows from (A)$^{\#}$ that $\beta^{\#}$ is monic. Moreover, $\alpha \circ \beta = i^{\bullet}$ in $\mathcal{S}_{\mathbf{T}}$ since this is precisely the content of (B). Thus if it can be shown that $\beta^{\#}$ is surjective, β will be the desired isomorphism in $\mathcal{S}_{\mathbf{T}}$ of the arbitrary summand α with a standard summand i. To this end, as ψ^{-1} is standard-deterministic, there exists a standard coproduct k, k' and γ, γ' as shown in the diagram

$$\begin{tikzcd} Q \arrow[r, "k"] \arrow[d, "\gamma"'] \arrow[dr, phantom, "(C)"] & P+P' \arrow[d, "\psi^{-1}"] & Q' \arrow[l, "k'"'] \arrow[d, "\gamma'"] \\ TR \arrow[r, "Ti"'] & TX & TR' \arrow[l, "Ti'"] \end{tikzcd}$$

We use this to show that the following triangle commutes in $\mathcal{S}_{\mathbf{T}}$:

$$\begin{tikzcd} Q \arrow[d, "\beta^{\#}\gamma"'] \arrow[dr, "k^{\bullet}"] & \\ P \arrow[r, "j^{\bullet}"'] & P+P' \end{tikzcd}$$

For

$$\begin{aligned} k^{\bullet} &= \eta_{P+P'} k = (\psi \circ \psi^{-1})k = \psi^{\#}\psi^{-1}k \\ &= \psi^{\#}(Ti)\gamma \text{ by (C)} \\ &= (Tj)\beta^{\#}\gamma \text{ by } (A)^{\#} \\ &= (j^{\bullet})^{\#}\beta^{\#}\gamma = j^{\bullet} \circ (\beta^{\#}\gamma). \end{aligned}$$

But then $\beta^{\#}\alpha$ is base by Lemma 7.13. It follows that, as subsets of $P + P'$, $Q \subset P$. By the same argument, $Q' \subset P'$ so $\beta^{\#}\gamma$ is a (base) isomorphism and $(\beta^{\#}\gamma)^{\#} = \beta^{\#}\gamma^{\#}$ is bijective so that $\beta^{\#}$ is surjective. We are done. □

9. Boolean monads

In this section we see that $\mathcal{S}_{\mathbf{T}}$ is a Boolean category for taut $\mathbf{T}$. This result was announced in [15].

Definition 9.1. Let $\mathcal{B}$ be a Boolean category. A monad $\mathbf{T}$ in $\mathcal{B}$ is a *Boolean monad* if the following conditions hold:
(1) $\mathbf{T}$ is nontrivial (so that $\mathcal{B} \to \mathcal{B}_{\mathbf{T}}$ is a subcategory).
(2) $\mathcal{B}_{\mathbf{T}}$ is a Boolean category.
(3) Summands are standard in $\mathcal{B}_{\mathbf{T}}$ (so that predicate transformers should map the same predicates, consistent with adding monads on the fly in functional programming).

It follows that, for a Boolean monad, $\mathcal{B} \to \mathcal{B}_{\mathbf{T}}$ is a strict Boolean functor.

We are at last ready to state and prove our second main result which is the equivalence of concepts (1) $\Leftrightarrow$ (2) below. The third condition is useful for establishing examples. We restrict to $\mathcal{S}$ because we do not yet know how to give a more general proof.

Theorem 9.2. *Let* $\mathbf{T}$ *be a monad in* $\mathcal{S}$. *Equivalent are*
(1) $\mathbf{T}$ *is taut*.
(2) $\mathbf{T}$ *is a Boolean monad*.
(3) $\mathbf{T}$ *is nontrivial*, *summands are standard in* $\mathcal{B}_{\mathbf{T}}$ *and for* i, j *summands below*, *if* (A) *is a pullback in* $\mathcal{B}$ *then* (B) *is a pullback in* $\mathcal{B}_{\mathbf{T}}$.

$$\begin{tikzcd} P \arrow[r, "\beta"] \arrow[d, "i"'] \arrow[dr, phantom, "\text{(A)}"] & TQ \arrow[d, "Tj"] \\ X \arrow[r, "\alpha"'] & TY \end{tikzcd} \qquad \begin{tikzcd} P \arrow[r, "\beta"] \arrow[d, "i^{\bullet}"'] \arrow[dr, phantom, "\text{(B)}"] & Q \arrow[d, "j^{\bullet}"] \\ X \arrow[r, "\alpha"'] & Y \end{tikzcd}$$

Proof. (1) $\Rightarrow$ (2) In Definition 9.1, (1) is given and (3) follows from Theorem 8.4. For $\alpha : X \to TY$ and $Q \subset Y$ construct pullbacks (C,C') as shown.

$$\begin{tikzcd} P \arrow[r, "i"] \arrow[d, "\beta"'] \arrow[dr, phantom, "(A)"] & X \arrow[d, "\alpha"] \arrow[dr, phantom, "(A')"] & X' \arrow[l, "i'"'] \arrow[d, "\beta'"] \\ TQ \arrow[r, "Tj"'] & TY & TQ' \arrow[l, "Tj'"] \end{tikzcd}$$

By the discussion following Definition 7.9 and Proposition 7.4(3), The following square is a pullback in $\mathcal{S}_{\mathbf{T}}$

$$\begin{tikzcd} P \arrow[r, "i^{\bullet}"] \arrow[d, "\beta"'] & X \arrow[d, "\alpha"] \\ Q \arrow[r, "j^{\bullet}"'] & Y \end{tikzcd}$$

which establishes the existence of $[\alpha]Q = P$. If α is a coproduct injection, it is monic (because summands are standard) so, because $\mathcal{S}$ is Boolean, the top row in (A, A') is a coproduct. It is now clear that $\mathbf{T}$ is a Boolean monad.

(2)⇒(3) Given the pullback (A) in $\mathcal{S}$ shown below

$$
\begin{tikzcd}
R \arrow[dr, "t"] \arrow[drr, bend left, "\gamma"] \arrow[ddr, "k"'] & & \\
& P \arrow[r, "\alpha"] \arrow[d, "i"'] \arrow[dr, phantom, "(A)"] & TQ \arrow[d, "Tj"] \\
& X \arrow[r, "\beta"'] & TY
\end{tikzcd}
$$

we must establish that (B) shown below is a pullback in $\mathcal{S}_{\mathbf{T}}$.

$$
\begin{tikzcd}
R \arrow[drr, bend left, "\gamma"] \arrow[ddr, "k^\bullet"'] & & \\
& P \arrow[ul, "u"'] \arrow[r, "\alpha"] \arrow[d, "i^\bullet"'] \arrow[dr, phantom, "(B)"] & Q \arrow[d, "j^\bullet"] \\
& X \arrow[r, "\beta"'] & Y
\end{tikzcd}
$$

(B) commutes because

$$\beta \circ i^\bullet = \beta^\# \eta_X i = \beta i = (Tj)\alpha = (j^\bullet)^\# \alpha = j^\bullet \circ \alpha.$$

Let $R = [\beta]Q$ in $\mathcal{S}_{\mathbf{T}}$, it being possible to choose k as a standard summand by hypothesis. As shown above, the pullback property of (A) induces unique t. On the other hand, unique u in $\mathcal{S}_{\mathbf{T}}$ is induced because $[\beta]Q$ is a pullback in that category. But u is a base morphism by Lemma 7.13, so $P = R$ by uniqueness of pullbacks, and (B) is $[\beta]Q$ in $\mathcal{S}_{\mathbf{T}}$.

(3)⇒(1) Consider the pullback (A) in $\mathcal{S}$

$$
\begin{tikzcd}
P \arrow[r, "\alpha"] \arrow[d, "i"'] \arrow[dr, phantom, "(A)"] & TQ \arrow[d, "Tj"] \arrow[dr, phantom, "(A^\#)"] & TP \arrow[l, "\alpha^\#"'] \arrow[d, "Ti"] \\
X \arrow[r, "\beta"'] & TY & TX \arrow[l, "\beta^\#"]
\end{tikzcd}
$$

and the necessarily commutative square (A$^\#$). By the discussion following Definition 7.9, (F$^\#$) is a pullback. That η is taut is automatic in $\mathcal{S}$ (see [16, Proposition 2.3]). □

We conclude the paper by relating taut monad maps to Boolean functors.

Proposition 9.3. *Let* $\mathbf{S}$, $\mathbf{T}$ *be Boolean monads in* $\mathcal{S}$ *and let* $\lambda : \mathbf{S} \to \mathbf{T}$ *be a monad map with induced functor* $H : \mathcal{S}_{\mathbf{S}} \to \mathcal{S}_{\mathbf{T}}$. *Then* λ *is taut if and only if* H *is a strict Boolean functor*.

Proof. First consider

$$\begin{tikzcd}
[\beta]Q \arrow[r, "\alpha"] \arrow[d, "i^\bullet"'] \arrow[dr, phantom, "\text{(A)}"] & SQ \arrow[r, "\lambda_Q"] \arrow[d, "Sj"] \arrow[dr, phantom, "\text{(B)}"] & TQ \arrow[d, "Tj"] \\
SX \arrow[r, "\beta^\#"'] & SY \arrow[r, "\lambda_Y"'] & TY
\end{tikzcd}$$

If H is Boolean then (A,B) is a pullback when (A) is. In particular, choose $\beta = 1_{TY}$ so that $\beta^\#$ is split epic since $\beta^\# \eta_{TY} = 1$. It follows from elementary category theory that (B) is a pullback (see [16, Lemma 1.1(3)]), and this shows λ is taut. Conversely, assume λ is taut. $HX = \emptyset \Rightarrow X = \emptyset$ since H is the identity on objects. For any λ, H preserves finite coproducts. H preserves $[\alpha]Q$ since, in the diagram above, if (A), (B) are pullbacks, so is (A,B). $\square$

References

[1] A. Carboni, Bicategories of partial maps, Cahiers Topologie Géom. Différentielle 28 (1987) 111–126.
[2] J.R.B. Cockett, S. Lack, Restriction categories I: categories of partial maps, Theoret. Comput. Sci. 270 (2002) 223–259.
[3] P.L. Curien, A. Obtulowicz, Partiality and cartesian categories, Inform. and Comput. 80 (1989) 50–95.
[4] E.W. Dijkstra, A Discipline of Programming, Prentice-Hall, Englewood Cliffs, NJ, 1976.
[5] R.A. DiPaola, A. Heller, Dominical categories: recursion theory without elements, J. Symbolic Logic 52 (1987) 595–635.
[6] S. Eilenberg, Automata, Languages and Machines, Vol. A, Academic Press, New York, 1974.
[7] C.C. Elgot, Monadic computation and iterative algebraic theories, in: H.E. Rose, J.C. Shepherdson (Eds.), Proc. Logic Colloq. '73, North-Holland, Amsterdam, 1975, pp. 175–230.
[8] M.J. Fischer, R.E. Ladner, Propositional dynamic logic of regular programs, J. Comput. System Sci. 18 (1979) 286–294.
[9] H. Kleisli, Every standard construction is induced by a pair of adjoint functors, Proc. Amer. Math. Soc. 16 (1965) 544–546.
[10] D. Kozen, A representation theorem for *-free PDL, Lecture Notes in Computer Science, Vol. 85, Springer, Berlin, 1980, pp. 351–362.
[11] M.V. Lawson, Inverse Semigroups, the Theory of Partial Symmetries, World Scientific Publishing, Singapore, 1998.
[12] G. Longo, E. Moggi, Cartesian closed categories and partial morphisms for effective type structures, in: G. Kahn, D.B. McQueen, G. Plotkin (Eds.), Internat. Symp. Semantics of Data Types, Lecture Notes in Computer Science, Vol. 173, Springer, Berlin, 1984, pp. 235–255.
[13] E.G. Manes, Predicate Transformer Semantics, Cambridge University Press, Cambridge, 1992.
[14] E.G. Manes, Implementing collection classes with monads, Math. Structures Comput. Sci. 8 (1998) 231–276.
[15] E.G. Manes, Boolean monads for functional programming, in: N. Callaos, S. Long, M. Loutfi (Eds.), Proc. World Multiconference on Systemics, Cybernetics and Informatics ISAS/SCI 2001, Orlando, Florida, USA, XVII, 2001, pp. 189–193.
[16] E.G. Manes, Taut monads and *T0* spaces, Theoret. Comput. Sci. 275 (2002) 79–109.
[17] E.G. Manes, Monads of sets, in: M. Hazewinkel (Ed.), Handbook of Algebra, Vol. 3, Elsevier, Amsterdam, 2003, pp. 67–153.
[18] E.G. Manes, Guarded and banded semigroups, Semigroup Forum, to appear.
[19] E. Moggi, Partial morphisms in categories of effective objects, Inform. and Comput. 76 (1988) 250–277.
[20] E. Moggi, Notions of computation and monads, Inform. and Comput. 93 (1991) 55–92.
[21] P.S. Mulry, Monads and algebras in the semantics of partial data types, Theoret. Comput. Sci. 99 (1992) 141–155.
[22] V.R. Pratt, Semantical considerations on Floyd-Hoare Logic, in: Proc. 17th IEEE Symp. the Foundations of Computer Science, 1976, pp. 109–121.
[23] E. Robinson, G. Rosolini, Categories of partial maps, Inform. and Comput. 79 (1988) 95–130.
[24] P. Wadler, Comprehending monads, Math. Structures Comput. Sci. 2 (1992) 461–493.