Semantics for Computational Effects: from Global to Local

John Power [1]

Department of Computer Science, University of Bath, Bath, BA2 7AY, United Kingdom

Abstract

We give a general construct that extends denotational semantics for a global computational effect to yield denotational semantics for a corresponding local computational effect. Our leading example yields a construction of the usual denotational semantics for local state from that for global state. Given any Lawvere theory L, possibly countable and possibly enriched, modelling a specific computational effect, we first give a universal construction that extends L, hence the global operations and equations of a given effect, to incorporate worlds of arbitrary finite size. Then, making delicate use of the final comodel of L, we give a construct that uniformly allows us to model *block*, the universality of the final comodel yielding a universal property of the construct. We illustrate both the universal extension of L and the canonical construction of *block* by seeing how they work primarily for state but also for nondeterminism, timing, exceptions, and interactive I/O.

Key words: computational effects, Lawvere theory, indexed Lawvere theory, model, monad, global state, local state.

1 Introduction

Over recent years, largely in collaboration with Martin Hyland and especially Gordon Plotkin, I have been working on the semantics of computational effects, refining Eugenio Moggi's proposal to use the notion of strong monad:

Email address: `A.J.Power@bath.ac.uk` (John Power).

[1] This document in an output from the PMI2 Project funded by the UK Department for Innovation, Universities and Skills (DIUS) for the benefit of the Japanese Higher Education Sector and the UK Higher Education Sector. The views expressed are not necessarily those of DIUS, nor British Council.

for an overview, see [24], and for a key journal paper, see [8]. The central idea has been to refine Moggi's use of monads [18,19] to the use of countable, possibly enriched, Lawvere theories: see [29] for an exposition of the relationship between the two notions, with examples drawn from computational effects, and see [10] for the history of the relationship. The central achievement of the work has been to allow an elegant denotational semantic account of the various ways in which individual effects naturally combine in programming languages like ML [8], the two main such ways being by the sum of Lawvere theories and by their tensor product.

We have successfully investigated other natural questions that arise in the study of computational effects too, including questions of operational semantics [21,26] and logics associated with effects [25]. But one question on which we have made a start but not yet fully resolved is that of giving denotational semantics for the process of extending from global computational effects to local computational effects [22,23]. That is the question we address here.

In order to model the usual global computational effects, with leading example being that of global state, one typically starts with a base category such as Set or ωCpo, then, using either a monad or a countable Lawvere theory, extends the category to model the effect at hand. For instance, the monad on Set for global state is $T_S - = (- \times S)^S$, where S is a set of states, typically given by V^{Loc} for a countable set V of values and a finite set Loc of locations. One can canonically model the computational λ-calculus together with operations for global state in the Kleisli category $Kl(T_S)$.

Expressing this in terms of Lawvere theories, the countable Lawvere theory L_S generating T_S is the theory freely generated by operations $lookup : V \longrightarrow Loc$ and $update : 1 \longrightarrow Loc \times V$ subject to computationally natural equations listed in [22]. Observe that the concept of Lawvere theory is purely syntactic, in that, unlike the monad, it does not depend upon an a priori choice of base category. One can duly model the computational λ-calculus together with operations for global state in a canonically determined subcategory of $Mod(L_S, Set)$ [30]. This is representative of the situation for all computational effects other than continuations.

In order to extend from global to local state, one replaces the base category, typically Set or ωCpo, by the functor category $[Inj, Set]$ or $[Inj, \omega Cpo]$, where Inj is the category of finite sets and injections [20,22,23], or equivalently, the category of natural numbers and monomorphisms. The monad for local state, on $[Inj, Set]$, is

$$(T_{LS}X)_n = \left(\int^{m \epsilon (n/Inj)} (Sm \times X_m) \right)^{Sn}$$

where $\int$ denotes a coend, a sophisticated kind of colimit given by a universal dinatural map [11,16], and where $Sn = V^n$ for a given set of values V. The idea is that in a state with n locations, a computation can create $m - n$ new locations and return a value, e.g., a function, that depends on them, cf [14,15]. In the case $V = 1$ this reduces to the monad for local names in [32].

In [22], with an oversight corrected in [23], we gave a sort of algebraic signature and set of equations that generated the monad T_{LS} for local state. Importantly for us, the signature and equations extended those for global state. But it had two drawbacks. First, it used a combination of cartesian and symmetric monoidal structure on $[Inj, Set]$, so was neither a Lawvere theory nor a symmetric monoidal Lawvere theory, but rather an uncomfortable and uninvestigated hybrid. And second, it applied only to state: the axioms related *block* with *lookup* and *update*, but with no general construct showing how to combine *block* with an arbitrary global signature, then instantiating that in the case of global state. So here, we refine the analysis of [22,23] by giving a construction that extends any Lawvere theory to account for localness, doing so without using a hybrid of cartesian and symmetric monoidal structure. The process uncovers sophisticated denotational structure underlying localness.

In order to extend global structure to local structure, one must first extend a Lawvere theory L to an indexed version of the same theory, i.e., if L has an operation $f : a \longrightarrow b$, then a local version also needs to have such an operation, but it must have it at each world, and it must be modified to account for the size of the world. For instance, for state, if one has n locations, there are n possible places at which one may either lookup or update. So we first seek a general construct that extends the operations of a global computational effect to become operations of an induced local effect, and that construct must be sensitive to the size of each world. The canonical symmetric monoidal structure of *Law* allows us to do that: Inj is the free symmetric monoidal category on 1 for which the unit is the initial object; *Law* is another symmetric monoidal category with unit the initial object; and so the universal property allows us to extend any Lawvere theory L to an Inj-indexed such theory $L_{\otimes}$, thus uniformly allowing us to extend global operations to local operations. The details are in Sections 2, 3, and 4.

Having extended the operations of a global effect to a local effect, one also needs to add an operation *block*: the central fact of localness is the capacity to create a new world. So we further need a general construct that builds *block* on top of any global effect. In the case of state, the semantics of *block* is a map of the form

$$M(n+1) \longrightarrow M(n)^V$$

uniformly in n and subject to natural equations. In order to provide such a semantic construct canonically, we first need to describe a construct whose semantics extends a model $M(n)$ at world n to a model of the form $M(n)^V$ at world $n+1$. To do that, we observe that V is the final comodel of L_V, i.e., the terminal object of the category $Comod(L_V, Set)$, and that exponentiating by the comodel V, the set $M(n)^V$ has $n+1$ *lookup* and *update* structures on it, n of them because $M(n)$ has n such structures, with an additional one determined by the comodel structure of V. We therefore investigate model structure induced by exponentiation by a comodel in Section 5.

The coherence condition on *block* asserts that it assigns the $(n+1)$-st variable the structure on $M(n)^V$ induced by the comodel structure of V, while maintaining the assignments of the earlier variables. That condition, together with a symmetry condition, suffice to characterise *block* semantically, i.e., in terms of models. But the spirit of the paper is to seek a construct that extends a Lawvere theory L for a global effect to include a new operation, subject to appropriate equations, for which a model consists of a model M of $L_\otimes$ together with, for each n, a map of the form

$$M(n+1) \longrightarrow M(n)^V$$

subject to appropriate axioms. We therefore investigate *block* at the level of models and also at the level of theories in Section 6.

The analysis of *block* at the level of Lawvere theories is the central new ingredient making this paper the journal version of [28].

2 Symmetric Monoidal Structures on the Category of Lawvere Theories

We first recall the definitions of Lawvere theory, map of Lawvere theories, and model of a Lawvere theory. Note that the category Nat of natural numbers and all maps between them has strictly associative finite coproducts, given by the sum of natural numbers; and consequently the category Nat^{op} has strictly associative finite products.

Definition 1 *A* Lawvere theory *consists of a small category L with strictly associative finite products and an identity-on-objects strict finite-product preserving functor*

$$J : Nat^{op} \longrightarrow L$$

The Lawvere theory is typically denoted merely by L. A map *of Lawvere theories from L to L' is a functor from L to L' that commutes with the func-*

tors from Nat^{op}: it necessarily strictly preserves finite products. A model *of L in any category C with finite products is a finite product preserving functor $M : L \longrightarrow C$.*

The definition forces the objects of a Lawvere theory L to be exactly the natural numbers. The definition of model specifically does not require strict preservation of finite products. For any Lawvere theory L, the models of L together with all natural transformations forms a category $Mod(L, C)$. If C is locally presentable, the functor

$$ev_1 : Mod(L, C) \longrightarrow C$$

has a left adjoint, yielding a monad T on C. The category $Mod(L, C)$ is then coherently equivalent to the category T-Alg. Taking C to be Set, the monads thus arising from Lawvere theories are exactly the finitary monads on Set.

We denote the category of Lawvere theories by Law.

It is routine to generalise the definition of Lawvere theory to allow for countable arities rather than finite ones: one systematically replaces finiteness by countability, replacing Nat by the category $\aleph_1$, a skeleton of the category of countable (including the possibility of finite) sets and all functions between them. We denote the category of countable Lawvere theories by Law_c.

It is also routine to generalise the notion of Lawvere theory to that of enriched Lawvere theory, the case of primary interest in computing being enrichment in the cartesian closed category ωCpo, as explained in detail in [8].

For this paper, our leading example of a countable Lawvere theory is that for state as follows [22]:

Example 2 *Let V be a countable set of values, and let Loc be a finite set of locations. Put $S = V^{Loc}$, so S is the set of states with Loc locations taking values in V. The countable Lawvere theory L_S for state is freely generated by operations $l : V \longrightarrow Loc$ and $u : 1 \longrightarrow Loc \times V$ subject to equations expressible in terms of an equational theory as follows, where loc is allowed to run over elements of Loc and v is allowed to run over elements of V:*

(1) $l_{loc}(u_{loc,v}(x))_v = x$
(2) $l_{loc}(l_{loc}(t_{vv'})_v)_{v'} = l_{loc}(t_{vv})_v$
(3) $u_{loc,v}(u_{loc,v'}(x)) = u_{loc,v'}(x)$
(4) $u_{loc,v}(l_{loc}(t_{v'})_{v'}) = u_{loc,v}(t_v)$
(5) $l_{loc}(l_{loc'}(t_{vv'})_{v'})_v = l_{loc'}(l_{loc}(t_{vv'})_v)_{v'}$ *where* $loc \neq loc'$
(6) $u_{loc,v}(u_{loc',v'}(x)) = u_{loc',v'}(u_{loc,v}(x))$ *where* $loc \neq loc'$
(7) $u_{loc,v}(l_{loc'}(t_{v'})_{v'}) = l_{loc'}(u_{loc,v}(t_{v'}))_{v'}$ *where* $loc \neq loc'$.

These equations can equally be expressed in diagrammatic form [22]. The first four express the interactions between the lookup and update operations, while the final three assert independence of the operations when they rely upon different locations. The idea is that the Lawvere theory is generated by operations for lookup and update, subject to natural computational equations between derived operations. A detailed discussion of the situation is given in [22]. The forgetful functor from $Mod(L_S, Set)$ to Set induces the monad $T_S - = (- \times S)^S$ for global state on Set. So, as explained in [22], this tells us that if one models lookup and update subject to natural computational equations, one derives the monad for global state, rather than needing to take the latter as a primitive.

Other examples of interest to us are given by nondeterminism, timing, exceptions, and interactive I/O as follows [8]:

Example 3 *The Lawvere theory L_N for binary nondeterminism is the Lawvere theory freely generated by a binary operation $\vee : 2 \longrightarrow 1$ subject to equations for associativity, commutativity and idempotence, i.e., the Lawvere theory for a semilattice. The induced monad on Set is the finite non-empty subset monad, F^+.*

Example 4 *Given a commutative monoid M, let L_M denote the Lawvere theory for M-sets, i.e., sets X together with an M-action $m : M \times X \longrightarrow X$ that respect the monoid structure of M. The Lawvere theory L_M is freely generated by M maps from 1 to 1, subject to equations corresponding to those for the monoid M. Such Lawvere theories can be used to model timing [8,13]. The induced monad on Set is $M \times -$.*

Example 5 *The Lawvere theory L_E for a set E of exceptions is the free Lawvere theory generated by an operation $raise_e : 0 \longrightarrow 1$ for each e in E. The induced monad on Set is $T_E = - + E$.*

Example 6 *The countable Lawvere theory $L_{I/O}$ for interactive input/output is the free countable Lawvere theory generated by operations $read : I \longrightarrow 1$ and $write : 1 \longrightarrow O$, where I is a countable set of* inputs *and O of* outputs*. The monad for interactive input/output $T_{I/O}(X) = \mu Y.(O \times Y + Y^I + X)$ is induced by this countable Lawvere theory: $T_{I/O}(X)$ is the free Σ-algebra on X, where $\Sigma Y = O \times Y + Y^I$; an algebra for Σ consists of an O-indexed family of unary operations and an I-ary operation.*

The category *Law* of Lawvere theories, possibly countable and enriched, possibly even, in the enriched setting, restricting to *discrete* Lawvere theories [9], possesses at least three distinct symmetric monoidal structures that are relevant to the study of computational effects.

The first, and simplest, of these symmetric monoidal structures is the sum: one takes the sum $L + L'$ of a pair of Lawvere theories L and L' in the category

Law. This is the most usual way in which computational effects are combined, yielding the usual combination of exceptions with all other computational effects investigated to date, and yielding the usual combination of interactive I/O with most other computational effects. The sum and its relevance to computational effects is studied in detail in [8].

The second symmetric monoidal structure on *Law*, and the one of primary interest to us here, is the *tensor* of Lawvere theories, cf [3]. The tensor of Lawvere theories takes a pair of Lawvere theories L and L', takes all the operations and equations of each, and insists that each operation of L commutes with each operation of L'. More formally, it is defined as follows:

Definition 7 *Given Lawvere theories* L *and* L'*, the Lawvere theory* $L \otimes L'$ *is defined by the universal property of having maps of Lawvere theories from* L *and* L' *to* $L \otimes L'$*, with commutativity of all operations of* L *with respect to all operations of* L'*, i.e., given* $f : A \longrightarrow B$ *in* L *and* $f' : A' \longrightarrow B'$ *in* L'*, we demand commutativity of the diagram*

$$\begin{array}{ccc} A \times A' & \xrightarrow{A \times f'} & A \times B' \\ {\scriptstyle f \times A'} \big\downarrow & & \big\downarrow {\scriptstyle f \times B'} \\ B \times A' & \xrightarrow[B \times f']{} & B \times B' \end{array}$$

Tensor, like sum, extends without fuss to countable Lawvere theories, and the tensor in Law_c is how global state is combined with almost all other computational effects. It is studied in detail in [8] and is characterised as follows [8]:

Theorem 8 *Given Lawvere theories* L *and* L'*, the tensor product* $L \otimes L'$ *induces an equivalence of categories*

$$Mod(L \otimes L', Set) \simeq Mod(L, Mod(L', Set))$$

The tensor forms a symmetric monoidal structure on *Law*, with unit given by the initial object of *Law*, which is Nat^{op} together with the identity functor.

The reason that the tensor is of primary interest to us here is because the tensor is central to our analysis of state [22], and the relationship between global and local state has been much more thoroughly investigated than that between global and local versions of other computational effects, see for instance [20], the series of papers associated with it, and [22,23]. Technically, the tensor seems to play a foundational role too, but it is not clear yet whether that is fundamental or merely coincidental.

A third symmetric monoidal structure on *Law* that is relevant to computational effects is the two-sided distributive tensor. That involves taking the sum $L + L'$ of a pair of Lawvere theories, then demanding that each operation of L distribute over each operation of L' and conversely. That symmetric monoidal structure was investigated in [9] and is central to Matthew Hennessy's combination of internal and external nondeterminism as he used in modelling concurrency in [5]. To date, it has been the least investigated of the three symmetric monoidal structures we have cited, and we merely mention it in passing here. It is studied more fully in the context of discrete Lawvere theories in [9].

There are other symmetric monoidal structures on *Law* too: an obvious one is given by the product of Lawvere theories. But, at present, we are not aware of others of relevance to computational effects.

3 Tensor Powers

Our primary use of the various symmetric monoidal structures on *Law* or its variants in this paper is as a way to extend a Lawvere theory L to an n-fold version of it, modelling the idea of having n possible locations. That leads us to consider the tensor product of n copies of L with respect to whichever is the most appropriate symmetric monoidal structure on *Law* for a given effect. Our leading example arises from state and uses a corollary of one of the central results of [8] as follows:

Theorem 9 *[8] Let L_S be the countable Lawvere theory for global state set S as in Example 2, and let $L_{S'}$ be the countable Lawvere theory for global state set S'. Then the tensor product $L_S \otimes L_{S'}$ is the countable Lawvere theory for global state set $S \times S'$.*

Corollary 10 *Let L_V be the countable Lawvere theory for global state V, understood to be a set of values V with one location. Then the n-fold tensor product $L_V^{(n)}$ is the countable Lawvere theory for global state V^n, i.e., the Lawvere theory for global state with value set V and n locations.*

So, if one has n locations, the n-fold tensor product $L_V^{(n)}$ of L_V is a characterisation of those operations generated by the operations of global state, namely *lookup* and *update*, subject to their equations.

Theorem 11 *Let L_N be the Lawvere theory for binary nondeterminism as in*

Example 3. Then $L_N \otimes L_N$ is itself L_N.

PROOF. This is essentially an example of the Eckmann-Hilton argument [2], which goes as follows:

let $(M, \circ, e)$ and $(M, \circ', e')$ be monoids whose operations commute with each other. Observe that $e = e'$ because the map $e : 1 \longrightarrow M$ respects the units of 1 and $(M, \circ', e')$, and the unit of 1 is the identity. Next observe that commutativity of $\circ$ with $\circ'$, together with the equality of the two units, implies that $x \circ y = y \circ' x$. Finally, use the same commutativity again but with units in different places to deduce that $x \circ y = x \circ' y$.

In our situation, a model of L_N in Set is a commutative and idempotent monoid. So it follows immediately from the Eckmann-Hilton argument that a model of L_N in $Mod(L_N, Set)$ is exactly a commutative idempotent monoid, as the two underlying monoid structures must agree and commutativity and idempotence are simply properties of the agreed monoid structure. And so $L_N \otimes L_N$ is L_N.

Corollary 12 *Let L_N be the Lawvere theory for binary nondeterminism. Then the n-fold tensor product $L_N^{(n)}$ is itself L_N.*

A priori, Corollary 12 may seem like a negative result in regard to the framework we propose here. But, as Eugenio Moggi pointed out at $MFPS$2006, one could argue that it is a positive result: in computing practice, nondeterminism is inherently global, so one should not expect it to vary in accordance with the number of possible locations, but rather remain the same, as it does here. We duly claim that the example of nondeterminism is a positive example of our framework.

Theorem 13 *[8] Let L_M be the Lawvere theory for M-sets for a commutative monoid M, as in Example 4; similarly for $L_{M'}$. Then the tensor product $L_M \otimes L_{M'}$ is the Lawvere theory for $(M \times M')$-sets.*

PROOF. Consider a model of L_M in $Mod(L_{M'}, Set)$. It consists of a set X together with an M-set structure m and an M'-set structure m' such that the square

$$\begin{array}{ccc} M \times M' \times X & \xrightarrow{M \times m'} & M \times X \\ {\scriptstyle (M' \times m) \circ (s \times X)} \downarrow & & \downarrow {\scriptstyle m} \\ M' \times X & \xrightarrow[m']{} & X \end{array}$$

commutes, where s is the symmetry. Symmetry supports a canonical distributive law of the monad $M \times -$ over the monad $M' \times -$, and commutativity of the square is the coherence condition for a lifting of algebras. So to give a model of L_M in $Mod(L_{M'}, Set)$ is equivalent to giving an $(M \times M')$-algebra, i.e., a model of $L_{M \times M'}$ in Set. So $L_M \otimes L_{M'}$ is $L_{M \otimes M'}$.

Corollary 14 *Let M be a commutative monoid and let L_M be the Lawvere theory for M-sets as used for timing as in Example 4. Then the n-fold tensor product $L_M^{(n)}$ is L_{M^n}, the Lawvere theory for M^n-sets.*

The tensor product allows us naturally to extend from having one clock with timing mechanism given by the commutative monoid M to having n clocks all with timing given by M. So tensor seems to be the natural way, or at least a natural way, to consider multiple copies of state or nondeterministic operators or clocks. But for exceptions and interactive I/O, sum seems to be a more natural combination.

Proposition 15 *Let L_E be the Lawvere theory for E exceptions as in Example 5, and let $L_{E'}$ be the Lawvere theory for E' exceptions. Then the sum $L_E + L_{E'}$ of Lawvere theories is the Lawvere theory $L_{E+E'}$ for $E+E'$ exceptions.*

PROOF. L_E is the free Lawvere theory on E maps from 0 to 1. The sum of Lawvere theories takes all the operations of each theory subject to the equations of each theory. So the sum $L_E + L_{E'}$ immediately yields the result.

Corollary 16 *Let L_1 be the Lawvere theory for one exception. Then the n-fold sum of copies of L_1 is the Lawvere theory for n exceptions.*

For interactive I/O, there are choices to be made, depending upon exactly how one has in mind interactive I/O on two locations working. For instance, does one have in mind writing at two locations simultaneously, or does one choose a location and write there? An answer to that question depends upon the specific computational phenomenon one has in mind. One might even ask whether the question is well-posed: under some circumstances, one might consider two collection of locations, one from which one may read and one to which one may write. Here, we shall analyse one possibility by means of the sum.

Proposition 17 *Let $L_{I/O}$ be the countable Lawvere theory for interactive I/O with countable input set I and countable output set O as in Example 6; similarly for $L_{I'/O'}$. Then the countable Lawvere theory $L_{I/O} + L_{I'/O'}$ is the countable Lawvere theory for interactively choosing a location and reading from it and choosing a location and writing to it.*

PROOF. $L_{I/O}$ is the free countable Lawvere theory generated by operations $read : I \longrightarrow 1$ and $write : 1 \longrightarrow O$ subject to no equations. Taking the sum with $L_{I'/O'}$ simply takes all the operations of both theories subject to no equations, hence the result.

Note that $L_{I/O}$ itself is the sum of the countable Lawvere theory L_I for inputting with the countable Lawvere theory L_O for outputting.

Corollary 18 *Let $L_{I/O}$ be the countable Lawvere theory for interactive I/O with input set I and output set O. Then the n-fold sum of copies of $L_{I/O}$ is the countable Lawvere theory for interactively choosing among n locations and reading from it and choosing among n locations and writing to it.*

4 Universal Structure of the Category of Natural Numbers and Injections

The category Inj of natural numbers and monomorphisms, equivalently the category of finite sets and injections, does not have finite coproducts. In particular, because of the absence of codiagonals, the sum of two natural numbers does not act as their binary coproduct in the category Inj. But Inj does have an initial object and, together with the sum of natural numbers, that equips Inj with a symmetric monoidal structure for which the unit is the initial object. It is elementary to verify that it is the free symmetric monoidal category on 1 with unit the initial object.

In Section 2, we saw that the tensor product of Lawvere theories, possibly countable or enriched, makes Law into a symmetric monoidal category with unit the initial object. So we can use the universal property of Inj in the light of Corollaries 10, 12 and 14.

The sum of Lawvere theories is also, trivially, a symmetric monoidal structure on Law with unit the initial object. So we can equally use the universal property of Inj in the light of Corollaries 16 and 18.

The universal property of Inj allows us, starting with an arbitrary Lawvere theory L, and using either the tensor or the sum of Lawvere theories, to generate a functor

$$Inj \longrightarrow Law$$

We can generalise the latter data to consider, for any small category D, a functor

$$D \longrightarrow Law$$

regard that notion as a generalised form of Lawvere theory, analyse it as such, then restrict again to the particular case of Inj to see whether we can derive some account of a local version of a computational effect from a global computational effect. As we shall see in later sections, a variant of this, replacing Inj by Inj_2, the free symmetric monoidal category on the arrow category with unit the initial object, together with a more refined notion of model, will also allow us to model *block* too.

Definition 19 *For any small category D, a D-*indexed Lawvere theory* is a functor $L : D \longrightarrow Law$. A* model *$M$ of a D-indexed Lawvere theory L in a category C with finite products consists of, for every object d of D, a model*

$$M_d : L_d \longrightarrow C$$

of L_d in C, together with, for each map $f : d \longrightarrow d'$ in D, a natural transformation

$$\begin{array}{ccc} L_d & \xrightarrow{M_d} & C \\ {\scriptstyle L_f}\downarrow & \Downarrow M_f & \downarrow{\scriptstyle Id} \\ L_{d'} & \xrightarrow[M_{d'}]{} & C \end{array}$$

respecting the composition and identities of D.The latter means that M_{id} is the identity and, given maps $f : d \longrightarrow d'$ and $f' : d' \longrightarrow d''$, the natural transformation $M_{f'f}$ is given by pasting $M_{f'}$ on to M_f.

It is straightforward to extend the set of models of a D-indexed Lawvere theory L in a category C with finite products to give a category $Mod(L, C)$: the arrows are D-indexed families of natural transformations that are coherent with respect to the M_f's. And it is routine to extend the definition and analysis of indexed Lawvere theories to account for countability and for enrichment, e.g., in the category ωCpo.

Corollaries 10, 12, 14, 16 and 18 immediately yield examples of Inj-indexed Lawvere theories and their models. The various examples are similar, so we shall only spell out the details for state, which is our leading example.

Example 20 *By Corollary 10, if we let L_V be the countable Lawvere theory for global state with value set V and one location, the n-fold tensor $L_V^{(n)}$ is the Lawvere theory for global state with value set V and n locations. There are $n+1$ canonical coprojections from $L_V^{(n)}$ to $L_V^{(n+1)}$ determined by the universal property of tensor as determined by Definition 7. Putting $D = Inj$, the universal property of Inj extends L_V to the functor*

$$(L_V)_\otimes : Inj \longrightarrow Law$$

sending n to $L_V^{(n)}$, with behaviour of $(L_V)_\otimes$ on maps determined by the canonical coprojections. A model M of $(L_V)_\otimes$ in a category C with finite products consists of, for each n, a model M_n of $L_V^{(n)}$, i.e., a model of the Lawvere theory for state with value set V and n locations, in C, respecting the structural inclusions of L_V into each $L_V^{(n)}$.

For any D-indexed Lawvere theory L and category C with finite products, there is a forgetful functor

$$U : Mod(L, C) \longrightarrow [D, C]$$

taking a model M to the functor sending d to $M_d(1)$. For each d, recalling that L_d is an ordinary Lawvere theory, the functor U has a d-component of the form

$$U_d : Mod(L_d, C) \longrightarrow C$$

For any locally presentable category C, each U_d has a left adjoint F_d, exhibiting $Mod(L_d, C)$ as monadic over C. These left adjoints and this monadicity result can be combined as follows:

Proposition 21 *For any D-indexed Lawvere theory L and any locally presentable category C, the forgetful functor $U : Mod(L, C) \longrightarrow [D, C]$ has a left adjoint F given pointwise, i.e., for any functor $X : D \longrightarrow C$,*

$$(FX)_d = F_d X_d$$

Moreover, the adjunction exhibits $Mod(L, C)$ as monadic over $[D, C]$.

PROOF. It follows from the study of ordinary Lawvere theories that an adjoint F_d exists for each d. One can routinely verify that such adjoints collectively satisfy the adjointness property for the forgetful functor U. Monadicity can be proved in several ways, one such proof being by Beck's monadicity theorem.

Our leading class of examples of D-indexed Lawvere theories for this section has $D = Inj$ and arises as follows:

Definition 22 *Given any ordinary Lawvere theory L, denote by $L_\otimes$ the Inj-indexed Lawvere theory freely generated by the tensor structure of Law and the fact of Inj being the free symmetric monoidal category on 1 with unit the initial object. So $(L_\otimes)n = L^{(n)}$, the n-fold tensor product of L using the tensor of Law.*

Applying Proposition 21 to the Inj-indexed Lawvere theories that thus arise allows us to conclude the following:

Corollary 23 *Given any ordinary Lawvere theory L and any locally presentable category C, let T be the monad on C induced by the forgetful functor*

$$Mod(L, C) \longrightarrow C$$

Then, the forgetful functor

$$Mod(L_\otimes, C) \longrightarrow [Inj, C]$$

exhibits $Mod(L_\otimes, C)$ as monadic over $[Inj, C]$ with monad $T_\otimes$ given by

$$(T_\otimes X)_n = T^{(n)} X_n$$

where $T^{(n)}$ is the monad on C induced by $L^{(n)}$.

Consider the significance of Corollary 23 to the example of global state. Applied to Example 20, it allows us to extend and reformulate Example 2 as follows:

Example 24 *The category $[Inj, Set]$ is a cartesian closed category. Given a set V of values, let V also denote the constant functor from Inj to Set at the object V of Set. Let Loc denote the object of $[Inj, Set]$ given by the inclusion of Inj into Set. In [22], using these conventions, we gave an enriched version of Example 2, with enrichment in the cartesian closed category $[Inj, Set]$. One freely has operations $l : V \longrightarrow Loc$ and $u : 1 \longrightarrow Loc \times V$ subject to enriched* *versions of the equations listed in Example 2. Denote this $[Inj, Set]$-enriched countable Lawvere theory by L_S. Then the category $Mod(L_S, [Inj, Set])$ of models of L_S in $[Inj, Set]$ is equivalent to the category $Mod((L_V)_\otimes, Set)$. By Corollary 23, the induced monad T on $[Inj, Set]$ is given by*

$$(TX)_n = T_{V^n} X_n$$

i..e, the value of TX at n is given by applying the monad for global state with value set V and n locations to the set X_n. Although they are equivalent categories, it is considerably easier to calculate with $Mod((L_V)_\otimes, Set)$ than with $Mod(L_S, [Inj, Set])$ as done in [22]. For, as in Example 20, an object of $Mod((L_V)_\otimes, Set)$ consists of a model M_n of $L_V^{(n)}$ for each n such that the maps in Inj are sent to structure-respecting functions. In contrast, calculation with $Mod(L_S, [Inj, Set])$ involves the exponential in $[Inj, Set]$, which is complicated.

The significance of this example is that it includes all operations and all equations that extend from global state as modelled in *Set* to $[Inj, Set]$. The only other operation one needs in order to model local state is a *block* operation subject to two sorts of equations: two equations only involving *block*, and equations relating *block* with the operations that exist globally. So, later in the paper, we shall account for *block*. But for the present, we return to other examples.

Example 25 *Let L_N be the Lawvere theory for binary nondeterminism as in Example 3. By Corollary 12, $(L_N)_\otimes n = L_N$. So $Mod((L_N)_\otimes, Set)$ is the category of models of L_N in $[Inj, Set]$, i.e., the category of semilattices in $[Inj, Set]$. The monad T induced by Corollary 23 on $[Inj, Set]$ is thus given by putting $(TX)_n$ equal to the set of finite non-empty subsets of X_n.*

Example 26 *Let L_M be the Lawvere theory for M-sets for a commutative monoid M as in Example 4. By Corollary 14, the n-fold tensor of L_M is L_{M^n}, the Lawvere theory for M^n-sets, accounting for timing with n clocks that measure time in terms of elements of M. So $(L_M)_\otimes n = L_{M^n}$. A model of $(L_M)_\otimes$ in Set consists of an M^n-set for each n, respecting the structural inclusions. The induced monad T on $[Inj, Set]$ is given by $(TX)_n = M^n \times X_n$.*

For exceptions, we replace the tensor in *Law* by the sum in *Law*. The universal property of *Inj* applies equally to the sum as it does to the tensor. Emulating Definition 22, given any Lawvere theory L, we let L_+ denote the *Inj*-indexed Lawvere theory freely generated by the sum of *Law*. So $(L_+)n$ is the sum of n copies of L. Proposition 21 again determines a monad on $[Inj, C]$ for any locally presentable category C.

Example 27 *Let L_1 be the Lawvere theory for one exception as in Exam-*

ple 5. By Corollary 16, the n-fold sum of L_1 is the Lawvere theory L_n for n exceptions. So $(L_1)_+n = L_n$. The induced monad T on $[Inj, Set]$ is given by $(TX)_n = X_n + n$, i.e., $- + J$, where J is the inclusion of Inj into Set.

There seems to be no single best way to extend interactive I/O: it depends upon exactly what one means by considering a number of possible locations. One possibility involves the replacement of Inj by $Inj \times Inj$, reflecting the idea of having separate locations from which one reads and to which one writes. One can readily consider maps from 2 into Law_c determined by taking the sum of the countable Lawvere theory L_I for inputting from I with that, L_O, for outputting to O, the sum being $L_{I/O}$. But in the simpler case of Inj, we have the following.

Example 28 *Let $L_{I/O}$ be the countable Lawvere theory for interactive I/O with input set I and output set O as in Example 6. By Corollary 18, the n-fold sum of $L_{I/O}$ is the Lawvere theory for selecting one of n locations and reading from it interactively with selecting one of n locations and writing to it, with input set I and output set O. The induced monad T on $[Inj, Set]$ is given by $(TX)_n = \mu Y.((n \times O \times Y) + (n \times Y^I) + X_n)$.*

For all of this discussion of this section, one could replace Inj by other categories of worlds, the key requirement for the theory we have outlined being that the relevant category of worlds is given by the free category on 1 with structure that is possessed by Law. In due course, this flexibility may allow the incorporation of current work by Hermida and Tennent, who are considering universal structure on various of the categories used by O'Hearn and Tennent in [20] in their modelling of local state [6].

5 Exponentiation by a Comodel

The technical development of the paper so far allows us to extend the operations and equations of a global effect to a local effect. But in order to account for localness, we also need to introduce a *block* operation. In order to do that, we need a general construction that extends an arbitrary model of the n-fold tensor product $L^{(n)}$ of a Lawvere theory L to a model of $L^{(n+1)}$: for *block* makes an assignment of structure to a new variable. In the rest of the paper, we investigate a canonical such construction: the central idea is that exponentiation by a comodel of a Lawvere theory yields a model of the theory.

Definition 29 *A* comodel *of a Lawvere theory L in a category C with finite coproducts is a model of L in C^{op}.*

Comodels and natural transformations form a category $Comod(L, C)$, with a canonical forgetful functor to C determined by evaluation at 1.

Theorem 30 *[31] For any Lawvere theory L, the forgetful functor*

$$U : Comod(L, Set) \longrightarrow Set$$

has a right adjoint, exhibiting $Comod(L, Set)$ as comonadic over Set.

PROOF. The assertion of the existence of a right adjoint is equivalent to the assertion that the inclusion of $Comod(L, Set)$ into $[L^{op}, Set]$ has a right adjoint. For that, the key point is that for any natural number n, the functor $n \times - : Set \longrightarrow Set$ sends a set X to the n-fold coproduct of copies of X, and it preserves limits of ω-cochains.

Given a functor $H : L^{op} \longrightarrow Set$, there is a canonical map $\phi_n : n \times H1 \longrightarrow H(n \times 1)$. A map in L of the form $f : n \longrightarrow 1$ is sent by H to a function of the form $Hf : H1 \longrightarrow H(n \times 1)$. So we need to pullback Hf along ϕ_n. We need to do that simultaneously for all maps in L, yielding a limit $H_1 1$ together with a function $\sigma_1 : H_1 1 \longrightarrow H1$. In general, H_1 will not extend to a finite-coproduct preserving functor because of non-triviality of

$$n \times \sigma_1 : n \times H_1 1 \longrightarrow n \times H1$$

But one can continue inductively to build a cochain of length ω: in defining $H_2 1$, one must also account for equalities in L, i.e., given $f : n \longrightarrow 1$ and $g_i : n_i \longrightarrow 1$ for $1 \leq i \leq n$, one must account for $f(g_1, \cdots, g_n)$. But after defining $H_2 1$, it is routine to extend to all n. In the limit, H_ω is functorial as one can define $H_\omega f$ by using the limiting property of the cochain given by $n \times \sigma_k$. It is routine to verify that H_ω is the right adjoint.

For comonadicity, it is routine to verify the conditions for the dual of Beck's monadicity theorem, i.e., that U reflects isomorphisms and that split equalisers lift [1].

Observe that the central point used here was cartesian closedness of Set, making enrichment in a cartesian closed category routine: we used the fact that the tensor of n with X had the universal property of a product. And the result generalises routinely to a countable version. One can duly generalise Theorem 30 to categories such as $Poset$ and ωCpo.

Corollary 31 *For any Lawvere theory L, the category $Comod(L, Set)$ has a terminal object, the final comodel.*

Our leading example, which is also the leading example of [31], is given by global state:

Example 32 *Let L_S be the countable Lawvere theory for global state, with $S = V^{Loc}$ and where Loc is finite. The induced comonad on Set is given by $(-)^S \times S$, and the final comodel is therefore given by $S = V^{Loc}$. A fortiori, the final comodel of L_V in Set is given by V, with structural maps given by*

$$loc = \delta : V \longrightarrow V \times V$$

and

$$upd = \pi_2 = t \times V : V \times V \longrightarrow 1 \times V \cong V$$

One can, in general, see the category of comodels for the Lawvere theory L_S as a category of arrays [31].

The category of comodels of a Lawvere theory is often much less substantial than the category of models.

Example 33 *Let L_N be the Lawvere theory for binary nondeterminism, as in Example 3. The category of models of L_N in Set is the category of semilattices, but the category of comodels is the trivial one-object category, the only comodel being given by the trivial comodel structure of the empty set. The final comodel is therefore trivially given by the empty set.*

Eugenio Moggi has pointed out that we could regard it as a positive statement about the framework developed here that the comodels of L_N are trivial: nondeterminism is inherently global, so one should positively want a theory of localness to yield a trivial outcome in the case of nondeterminism.

Example 34 *Let L_M be the Lawvere theory for M-sets for a monoid M as in Example 4 for modelling timing. A comodel for L_M in Set is exactly the same as a model, i.e., an M-set. So the the category of comodels is the category of M-sets, and the final comodel is the final M-set, which is* 1.

Example 35 *Let L_E be the Lawvere theory for E exceptions as in Example 5, and assume E is non-empty. There is just one comodel of L_E in Set, namely the empty set. So the final comodel is the empty set.*

Example 35 looks trivial, but there is a sense in which exceptions carry little structure, so it seems okay. Our analysis cannot yet model *handle* (but see [27]), so a development of our analysis that does model *handle* may well lead to something more substantial.

Example 36 *Let $L_{I/O}$ be the countable Lawvere theory for inputting from I and outputting O as in Example 6. A comodel in Set consists of a set X together with I unary operations on X and a function from $O \times X$ to X, subject to no equations. The latter function appears in the definition of a model of $L_{I/O}$ but with O replaced by I. The final comodel is* 1.

I am not sure what to make of Example 36: the category of comodels is non-trivial and reflects natural structures arising from interactive I/O, but the structure is unfamiliar and the final comodel is insubstantial.

Although the various examples other than state typically look simple, the final comodel does, as we shall see, play a fundamental role in local state, which is our leading example.

Given a comodel $A : L^{op} \longrightarrow Set$ of an arbitrary Lawvere theory L in Set and given an arbitrary object X of a category C with products, letting X^{-} denote a product of copies of X in C, the composite

$$L \xrightarrow{\quad A \quad} Set^{op} \xrightarrow{\quad X^{-} \quad} C$$

preserves finite products. Thus X^{A1}, the $A1$-fold product in C of copies of X, inherits an L-model structure in C from the L-comodel A in Set. More generally, for any Lawvere theory L' and any model $M : L' \longrightarrow C$ in a category C with products, the composite

$$L \xrightarrow{\quad A \quad} Set^{op} \xrightarrow{\ M^{-} \ } Mod(L', C)$$

preserves finite products. By Theorem 8, to give such a finite product preserving functor is equivalent to giving a model of $L' \otimes L$. This is natural in M, yielding the following:

Proposition 37 *For any Lawvere theory L and any comodel A of L in Set, and for any Lawvere theory L' and any category C with products, exponentiation $(-)^A$ induces a functor*

$$(-)^A : Mod(L', C) \longrightarrow Mod(L' \otimes L, C)$$

Corollary 38 *For any Lawvere theory L, let F be the final comodel of L in Set. Then, for any category C with products, exponentiation $(-)^F$ induces a functor*

$$(-)^F : Mod(L_{\otimes} n, C) \longrightarrow Mod(L_{\otimes}(n+1), C)$$

Moreover, this is natural in n as an object of Inj.

This duly generalises to the countable setting and also to the enriched setting. Moreover, as there is a canonical map of Lawvere theories from $L' + L$ to $L' \otimes L$ for any Lawvere theories L and L', and so a canonical map from $L_+(n+1)$ to $L_\otimes(n+1)$, Corollary 38 yields, for any Lawvere theory L and category C with products, a functor

$$Mod(L_+ n, C) \longrightarrow Mod(L_+(n+1), C)$$

So our axiomatic development applies not only to state, nondeterminism and timing, but also to exceptions and interactive I/O. Our leading example, Example 24, extends as follows:

Example 39 *Consider a model M of the Inj-indexed Lawvere theory $(L_V)_\otimes$ in Set. Each $(M_n 1)^V$ canonically possesses the structure of a model of $L_V^{(n+1)}$: it inherits n L_V-structures from those of $M_n 1$, while the final L_V-comodel structure of V given by Example 32 induces a further L_V-structure on $(M_n 1)^V$.*

The significance for us is that the semantics of $block_n$ will be a map of $L_V^{(n+1)}$-models from M_{n+1} to M_n^V.

Observe that the above discussion is inherently semantic, i.e., it is about constructs on models. But the spirit of the idea of Lawvere theories is that we seek constructs on theories, and then consider a notion of model of that in any well-behaved semantic category C such as Set. In order to provide a construct at the level of theories, observe that, for any comodel A of a Lawvere theory L, the functor

$$(-)^A : Mod(L', C) \longrightarrow Mod(L' \otimes L, C)$$

of Proposition 37 makes the following diagram commute:

$$\begin{array}{ccc} Mod(L', C) & \xrightarrow{(-)^A} & Mod(L' \otimes L, C) \\ {\scriptstyle ev_1}\downarrow & & \downarrow{\scriptstyle ev_1} \\ C & \xrightarrow[(-)^{A1}]{} & C \end{array}$$

Letting $C = Set$ for example, the functor $(-)^{A1} : C \longrightarrow C$ is monadic [17,26], with left adjoint given by $A1 \times -$ and hence with monad given by $(A1 \times -)^{A1}$, the monad for state with state set $A1$. So, letting L_{A1} denote the Lawvere theory for state with state set $A1$, it follows that $Mod(L', C)$ is equivalent to $Mod(L', Mod(L_{A1}, C))$, which is in turn equivalent to $Mod(L' \otimes L_{A1}, C)$, and the diagram

$$\begin{array}{ccc} Mod(L', C) & \longrightarrow & Mod(L' \otimes L, C) \\ \simeq \Big\uparrow & & \Big\downarrow ev_1 \\ Mod(L' \otimes L_{A1}, C) & \xrightarrow[ev_1]{} & C \end{array}$$

commutes. By a standard result about Lawvere theories [1], we can conclude the following:

Theorem 40 *For any Lawvere theory L and any comodel A of L in Set, and for any Lawvere theory L' and any locally presentable category C, there is a map of Lawvere theories $f : L' \otimes L \longrightarrow L' \otimes L_{A1}$ such that the functor*

$$Mod(L' \otimes L_{A1}, C) \xrightarrow{\simeq} Mod(L', C) \xrightarrow{(-)^A} Mod(L' \otimes L, C)$$

is of the form $Mod(f, C)$.

This can duly be iterated, yielding, for any n, a map of Lawvere theories of the form $L' \otimes L^{(n)} \longrightarrow L' \otimes L_{A1^n}$. We shall use this construct to elucidate *block* at the level of theories in Section 6.

Observe that putting $L' = Nat^{op}$ and $C = Set$, the functor

$$(-)^A : Set \longrightarrow Mod(L, Set)$$

of Proposition 37 has a left adjoint: that left adjoint agrees exactly with the central construct used in [26] to analyse structural operational semantics for computational effects.

6 Introducing Block

We have foreshadowed *block* in Section 5. In general, one starts with a Lawvere theory L. By Corollary 31, that Lawvere theory L must have a final comodel F. By Corollary 38, for any category C with products, exponentiation $(-)^F$ induces a functor

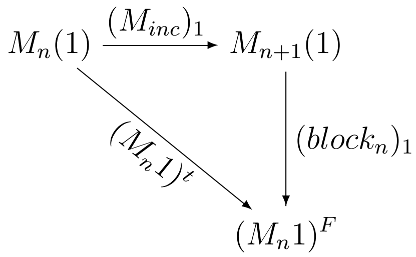
$$(-)^F : Mod(L_{\otimes} n, C) \longrightarrow Mod(L_{\otimes}(n+1), C)$$

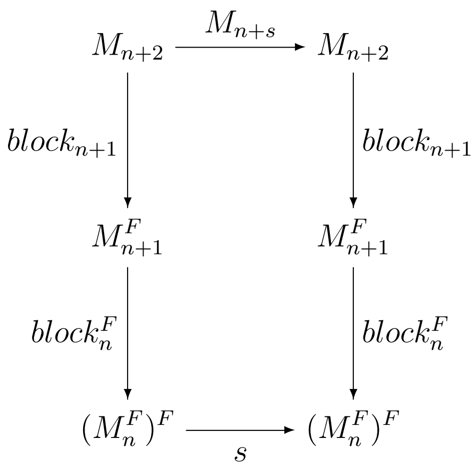
natural in n.

Definition 41 *Given an arbitrary Lawvere theory L and a category C with products, a $block$-*algebra *in C consists of a model M of the Inj-indexed Lawvere theory $L_{\otimes}$ in C, together with an indexed family of maps*

$$block_n : M_{n+1} \longrightarrow M_n^F$$

in $Mod(L_{\otimes}(n+1), C)$, natural with respect to Inj, satisfying the following two equations:





where the second occurrence of s means swapping the two copies of F respectively.

The first axiom asserts that *block* only affects the newly created location, while the second asserts that in creating two new locations and assigning structure to them, one can do so in either order providing one remembers which location is being assigned which structure.

One can routinely form a category of *block*-algebras, the maps being maps of models of $L_{\otimes}$ that respect the *block* structure. We denote the category by *block-Alg*.

Theorem 42 *If C is locally presentable, the composite forgetful functor*

$$block\text{-}Alg \longrightarrow [Inj, C]$$

is monadic.

PROOF. There are several proofs of this: a *block*-algebra is, by construction, given by adding operations subject to equations to a model of $L_\otimes$, and such models are in turn given by operations and equations on an object of $[Inj, C]$. Combining these facts, a *block*-algebra consists of an object of $[Inj, Set]$ together with operations subject to universally defined equations, and so the category of such is monadic over $[Inj, C]$ as an instance of the general theory of [12].

Definition 41 and Theorem 42 extend routinely from tensor to sum via the canonical map from a sum of Lawvere theories to its tensor. So, subject to that variation as appropriate, the result applies to all of our examples. We now check that, if L is the Lawvere theory for global state with value set V and one location, it yields the usual monad T_{LS} for local state on $[Inj, Set]$, following O'Hearn et al [14,15,20,32] and agreeing with the monad of [22,23].

The monad for local state, on $[Inj, Set]$, is

$$(T_{LS}X)_n = (\int^{m\epsilon(n/I)} (Sm \times X_m))^{Sn}$$

where $\int$ denotes a coend, a sophisticated kind of colimit given by a universal dinatural map [11,16], and where $Sn = V^n$ for a given set of values V, cf [14,15]. If $V = 1$ it reduces to the monad for local names in [32].

The behaviour of T_{LS} on injective maps $f : n \longrightarrow n'$ is as follows: decompose n' as the sum $n + n''$, note that $S(p + n'') = Sp \times Sn''$, and use covariance of X. So the map

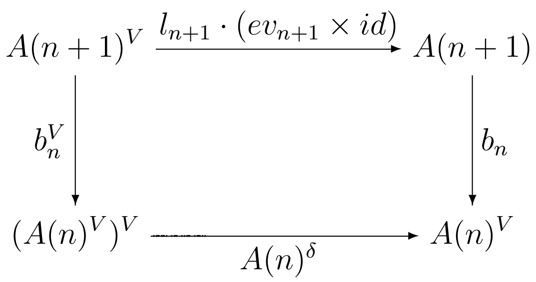
$$(\int^{m\epsilon(n/I)} (Sm \times X_m))^{Sn} \times Sn \times Sn'' \longrightarrow \int^{m''\epsilon((n+n'')/I)} (Sm'' \times X_{m''})$$

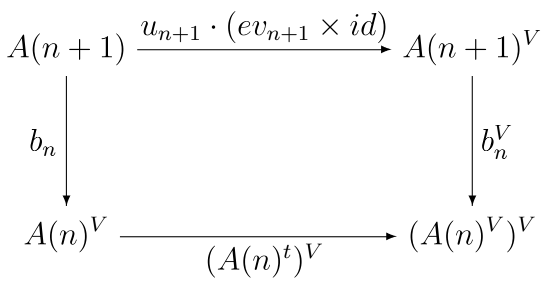
evaluates at Sn, then maps the m-th component of the first coend into the $(m+n'')$-th component of the second, using the above isomorphism for S and functoriality of X.

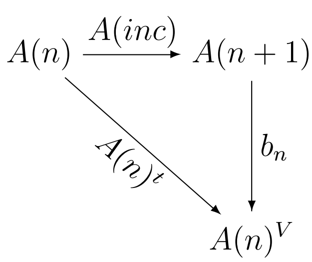
In [22,23], an algebraic signature generating T_{LS} was given by defining the corresponding category $LS([Inj, Set])$ of algebras as follows, subject to some not entirely trivial rewriting: an algebra consists of

- an object A of $[Inj, Set]$
- a lookup map with components $l_n : n \times A(n)^V \longrightarrow A(n)$
- an update map with components $u_n : n \times A(n) \longrightarrow A(n)^V$
- a block map with components $b_n : A(n+1) \longrightarrow A(n)^V$

subject to commutativity of seven interaction diagrams and six commutativity diagrams. The interaction diagrams consist of the four interaction diagrams for global state, together with







The commutation diagrams are those for global state together with

$$\begin{array}{ccc}
A(n+2) & \xrightarrow{A(n+s)} & A(n+2) \\
\downarrow b_{n+1} & & \downarrow b_{n+1} \\
A(n+1)^V & & A(n+1)^V \\
\downarrow b_n^V & & \downarrow b_n^V \\
(A(n)^V)^V & \xrightarrow[s]{} & (A(n)^V)^V
\end{array}$$

and, for k less than or equal to n,

$$\begin{array}{ccc}
A(n+1)^V & \xrightarrow{l_{n+1} \cdot (ev_k \times id)} & A(n+1) \\
\downarrow b_n^V & & \downarrow b_n \\
(A(n)^V)^V & \xrightarrow[l_n \cdot (ev_k \times id)]{} & A(n)^V
\end{array}$$

$$\begin{array}{ccc}
A(n+1) & \xrightarrow{u_{n+1} \cdot (ev_k \times id)} & A(n+1)^V \\
\downarrow b_n & & \downarrow b_n^V \\
A(n)^V & \xrightarrow[u_n \cdot (ev_k \times id)]{} & (A(n)^V)^V
\end{array}$$

One can observe directly the correspondence between these axioms and those we have developed: recalling that we refer to n copies of each of *lookup* and *update* rather than one n-parametrised version of each, we have already accounted for precisely the diagrams arising from global state in Example 24. So to give A together with l and u is equivalent to giving a model M of $(L_V)_\otimes$ in *Set*. Now recall, from Example 32, that V is the final comodel of L_V. So, using Example 39, the first two interaction axioms together with the last two commutation axioms are equivalent to the assertion that $b_n : M_{n+1} \longrightarrow M_n^V$

is a map in $Mod(L^{(n+1)}, Set)$. The remaining two axioms are exactly the axioms for *block* in Definition 41. This, together with the characterisation of T_{LS} in [22,23], allows us to conclude the following:

Theorem 43 *Taking L to be the countable Lawvere theory L_V for global state with value set V and one location, the category block-Alg is equivalent to $LS([Inj, Set])$ and is thus monadic over $[Inj, Set]$ with monad T_{LS} that for local state.*

Example 44 *Let L_N be the Lawvere theory for binary nondeterminism as in Example 3. By Corollary 12, $L_N^{(n)}$ is L_N. And by Example 33, the final comodel in Set is the empty set. So block is trivial, and the induced monad is the monad for semilattices on $[Inj, Set]$ as in Example 25.*

Example 45 *Let L_M be the Lawvere theory for M-sets for commutative monoid M as in Example 4. By Corollary 14, $L_M^{(n)}$ is L_{M^n}, the Lawvere theory for M^n-sets. By Example 34, the final comodel in Set is 1. It follows from Example 26 that the induced monad is*

$$(T_{LM}X)_m = \int^{m\epsilon(n/I)} (M^m \times X_m)$$

Example 46 *Let L_1 be the Lawvere theory for one exception. By Corollary 16, $(L_1)_+n$ is the Lawvere theory for n exceptions. By Example 35, the final comodel in Set is the empty set. So block is trivial and the induced monad on $[Inj, Set]$ is $-+J$, where J is the inclusion of Inj into Set, as in Example 27.*

Example 47 *. Let $L_{I/O}$ be the countable Lawvere theory for interactive I/O with input set I and output set O, as in Example 6. By Corollary 18, $(L_{I/O})_+n$ is the Lawvere theory for interactively choosing one of n locations and reading from a copy of I and choosing one of n locations and writing to a copy of O. By Example 36, the final comodel is 1. I do not yet have a characterisation of the induced monad on $[Inj, Set]$.*

We end by giving a Lawvere-style account of *block*.

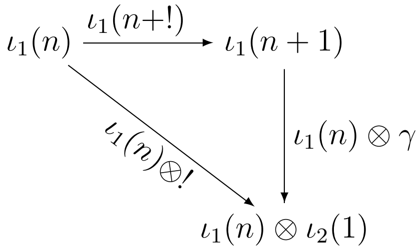
Definition 48 *Let Inj_2 denote the free strict symmetric monoidal category on the arrow category $\cdot \to \cdot$ with unit the initial object.*

For concreteness, assume that the generating arrow for Inj_2 is

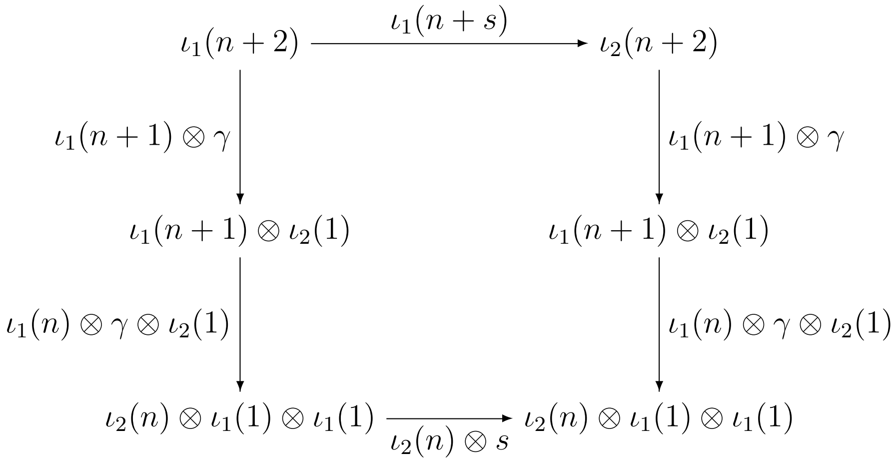
$$x \xrightarrow{\gamma} y$$

An arbitrary object of Inj_2 is a list of x's and y's, with the empty list acting as the unit. There are fully faithful functors from Inj into Inj_2, one of them, ι_1, sending n to a list of n copies of x, the other, ι_2, sending n to a list of n copies of y.

Observe that the following diagrams commute:



(1)



(2)

where the two occurrences of s mean the evident twist maps.

By Theorem 5, every Lawvere theory L together with a comodel A in Set generates a map of Lawvere theories $A^* : L \longrightarrow L_{A1}$ and hence an Inj_2-indexed Lawvere theory $L_{\otimes A}$. Note that $L_{\otimes A}(\iota_2(n)) = L_{A1^n}$, the Lawvere theory for state with state set $A1^n$. We need to consider some, but not all, of the models of the Inj_2-indexed Lawvere theory $L_{\otimes A}$.

Definition 49 *Given a Lawvere theory L together with a comodel A of L in Set, an* A-model *of the free Inj_2-indexed Lawvere theory $L_{\otimes A}$ on $A^* : L \longrightarrow L_{A1}$ in a category C with products is a model M of $L_{\otimes A}$ in C such that*

$$M_{z\otimes\iota_2(n)}- = (M_z-)^{A1^n} : L_{\otimes A}(z\otimes\iota_2(n)) \longrightarrow C$$

naturally in $z\epsilon Inj_2$ and strong symmetric monoidal naturally in $n\epsilon Inj$.

The definition of A-model extends routinely to yield a category $AMod(L, C)$, the arrows being the maps of models of $L_{\otimes A}$ that respect the A-structure.

Theorem 50 *If C is locally presentable, and F is the final coalgebra of a*

Lawvere theory L, the categories $FMod(L,C)$ and block-Alg are coherently equivalent.

PROOF. The proof is given by tedious checking of a mass of detail. The heart of it is the fact that the coherence conditions in Definition 49 mean that an A-model of the free Inj_2-indexed Lawvere theory $L_{\otimes A}$ on $A^* : L \longrightarrow L_{A1}$ in C is given by

- a model M of $L_{\otimes}$ in C
- for each n, a map of $L_{\otimes}(n+1)$-models

$$M_{n+1} \longrightarrow M_n^F$$

- subject to naturality in $n\epsilon Inj$
- subject to commutativity of the images of (1) and (2).

Considerable calculation shows that preservation of the commutativity of (1) and (2), together with the naturality conditions, yields preservation of all commutative diagrams in Inj_2, while agreeing with the two commutativity conditions in the definition of *block*-algebra.

Letting L be the Lawvere theory for globale state with one location, Theorems 43 and 50 yield both semantic and syntactic characterisations respectively of *block* for local state.

7 Conclusions and Further Work

Monads have been used to model aspects of computational effects since the late 1980's [18,19]. The equivalence between finitary monads and Lawvere theories was elaborated in the 1960's. So one might reasonably expect that it would have been clear from the outset that computational effects could be captured in either way. But, as a matter of historical fact, that is not true, as explained in [10].

The relationship between monads and Lawvere theories was an observation of no practical value in regard to computational effects until it was recognised, in 2001, that there is computationally natural presentation of the Lawvere theory for global state [22]. That recognition, followed quickly by corresponding recognition for all the other leading examples except for continuations, opened the way to the modelling of deeper aspects of computational effects that one cannot model in terms of the definition of monad. For example, monads do not model the operations that give rise to computational effects, such as *lookup* and *update*, whereas the usual presentations of Lawvere theories inherently involve operations, making it clear how to model them. For another example, the universal property of the tensor product of Lawvere theories is inexpressible in terms of monads. The following years duly saw a succession of papers developing those deeper aspects of computational effects, for instance [7–9,21,25,26,30].

In functional programming, one does not use semantics directly as one has syntax but not equations. Nevertheless, in informal reasoning, functional programmers implicitly use equations, for instance in their understanding of global state. Because they do not encode equations, they cannot simply discard monads in the light of the Lawvere theory work. But what they can do, but have not yet done, is to encode computational effects in terms of both monads and operations, then, in their informal reasoning, use the fact that the operations subject to the associated natural equations generate the monads just as they currently use the fact that the monads satisfy the monad axioms.

This specific paper is best understood as part of the general development. It develops the metatheory of Lawvere theories in order to provide a formal relationship between semantics for global state and semantics for local state, focusing on ease of calculation and a desire to model as directly as possible the ways in which programmers reason about local state: they do not study the complicated cartesian closed and linear structures of $[Inj, Set]$ but rather the relationship between a natural number n and its successor $n + 1$, so we have tried to draw that to the forefront here. This is not the only possible approach to modelling localness, see for instance [33].

The notion of *block* here deserves detailed comparison with the statically scoped block mechanism of Algol-like languages of [20]. In particular, it should relate to the subtle program equivalences investigated by O'Hearn and Tennent, continued by Hermida and Tennent [6]. Central to this paper are the universal properties of Inj and Inj_2. Universality is also central to Hermida and Tennent's work. So there seems reason to expect that there may be common ground.

It should further be possible to extend the work of this paper to the big-step structural operational semantics of Ghica [4]. It is intriguing that this paper makes fundamental use of the final comodel of a Lawvere theory in order to model *block*, while [26] makes fundamental use of the final comodel of a Lawvere theory in order to give a semantic foundation for structural operational semantics. That seems unlikely to be merely coincidental.

References

[1] Barr, M., and C. Wells, "Toposes, Triples, and Theories," Grundlehren der math. Wissenschaften **278**, Springer, 1985.

[2] Eckmann, B., and P.J. Hilton, *Group-like Structures in General Categories I: Multiplications and Comultiplications*, Math. Annalen **145** (1962), 227–255.

[3] Freyd, P.J., *Algebra-Valued Functors in General and Tensor Products in Particular*, Colloq. Math. Wroclaw **14** (1966) 89–106.

[4] Ghica, D., *Semantical Analysis of Specification Logic, 3: An Operational Approach*, "Proc. ESOP 2004," Lecture Notes in Computer Science **2986** (2004), 264–278.

[5] Hennessy, M.C.B., "Algebraic Theory of Processes," MIT Press, 1988.

[6] Hermida, C.A., and R.D. Tennent, *Monoidal Indeterminates and Categories of Possible Worlds*, draft.

[7] Hyland, J.M.E., P. Levy, G.D. Plotkin and John Power, *Combining Algebraic Effects with Continuations*, Theoretical Computer Science **375** (2007) 20–40.

[8] Hyland, J.M.E., G.D. Plotkin and John Power, *Combining Effects: Sum and Tensor*, Theoretical Computer Science **357** (2006), 70–99.

[9] Hyland, J.M.E., and John Power, *Discrete Lawvere Theories and Computational Effects*, Theoretical Computer Science **366** (2006) 144–162.

[10] Hyland, J.M.E., and John Power, *The Category Theoretic Understanding of Universal Algebra: Lawvere Theories and Monads*, Electronic Notes in Theoretical Computer Science **172** (2007) 437–458.

[11] Kelly, G.M., "Basic Concepts of Enriched Category Theory," Cambridge University Press, Cambridge, 1982.

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

[13] Kick, M., A.J. Power, and A.K. Simpson, *Coalgebraic Semantics for Timed Processes*, Information and Computation **204** (2006), 588–609.

[14] Levy, P.B., *Call-by-Push-Value: A Subsuming Paradigm*, "Proc. TLCA '99," Lecture Notes in Computer Science **1581** (1999), 228–242.

[15] Levy, P.B., "Call-by-push-value. A Functional/Imperative Synthesis," Semantic Structures in Computation, Springer, Berlin, 2004.

[16] Mac Lane, S., "Categories for the Working Mathematician," Springer, Berlin, 1971.

[17] Métayer, F., *State Monads and their Algebras*, arXiv:math/0407251v1, 2004.

[18] Moggi, E., *Computational Lambda-Calculus and Monads*, "Proc. LICS '89," IEEE Press (1989), 14–23.

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

[20] O'Hearn, P.W., and R.D. Tennent, "Algol-like Languages," Progress in Theoretical Computer Science, Birkhauser, Boston, 1997.

[21] Plotkin, G.D., and John Power, *Adequacy for Algebraic Effects*, "Proc. FOSSACS 2001," Lecture Notes in Computer Science **2030** (2001), 1–24.

[22] Plotkin, G.D., and John Power, *Notions of Computation Determine Monads*, "Proc. FOSSACS 2002," Lecture Notes in Computer Science **2303** (2002) 342–356.

[23] Plotkin, G.D., and John Power, *Erratum*, See the webpage http://homepages.inf.ed.ac.uk/gdp/publications/

[24] Plotkin, G.D., and John Power, *Computational Effects and Operations: an Overview*, "Proc. Domains 2002," Electronic Notes in Theoretical Computer Science **73** (2002), 149–163.

[25] Plotkin, G.D., and John Power, *L*ogic for Computational Effects: work in progress, "Proc. International Workshop on Formal Methods 03," British Computer Society Electronic Workshops in Computing (2003).

[26] Plotkin, G.D., and John Power, *T*ensors of Comodels and Models for Operational Semantics, "Proc. MFPS 2008," Electronic Notes in Theoretical Computer Science **218** (2008) 295–311.

[27] Plotkin, G.D., and M. Pretnar, *H*andlers of Algebraic Effects, "Proc. ESOP 2009," Lecture Notes in Computer Science **5502** (2009), 80–84.

[28] Power, John, *Semantics for Local Computational Effects*, "Proc. MFPS 2006," Electronic Notes in Theoretical Computer Science **158** (2006), 355–371.

[29] Power, John, *Countable Lawvere Theories and Computational Effects*, "Proc. MFCSIT 04," Electronic Notes in Theoretical Computer Science **161** (2006), 59–71.

[30] Power, John, *Generic Models for Computational Effects*, Theoretical Computer Science **364** (2006) 254–269.

[31] Power, John, and Olha Shkaravska, *From Models to Comodels: State and Arrays*, "Proc. CMCS 2004," Electronic Notes in Theoretical Computer Science **100** (2004), 185–200.

[32] Stark, I., *Categorical Models for Local Names*, Lisp and Symbolic Computation **9** (1996), 77–107.

[33] Staton, S., *Two Cotensors in One: Presentations of Algebraic Theories for Local State and Fresh Names*, "Proc. MFPS 2009," Electronic Notes in Theoretical Computer Science (to appear).