Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Electronic Notes in Theoretical Computer Science 172 (2007) 437–458
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

The Category Theoretic Understanding of Universal Algebra: Lawvere Theories and Monads

Martin Hyland[2]

Dept of Pure Mathematics and Mathematical Statistics
University of Cambridge
Cambridge, ENGLAND

John Power[1,3]

Laboratory for the Foundations of Computer Science
University of Edinburgh
Edinburgh, SCOTLAND

Abstract

Lawvere theories and monads have been the two main category theoretic formulations of universal algebra, Lawvere theories arising in 1963 and the connection with monads being established a few years later. Monads, although mathematically the less direct and less malleable formulation, rapidly gained precedence. A generation later, the definition of monad began to appear extensively in theoretical computer science in order to model computational effects, without reference to universal algebra. But since then, the relevance of universal algebra to computational effects has been recognised, leading to renewed prominence of the notion of Lawvere theory, now in a computational setting. This development has formed a major part of Gordon Plotkin's mature work, and we study its history here, in particular asking why Lawvere theories were eclipsed by monads in the 1960's, and how the renewed interest in them in a computer science setting might develop in future.

Keywords: Universal algebra, Lawvere theory, monad, computational effect.

1 Introduction

There have been two main category theoretic formulations of universal algebra. The earlier was by Bill Lawvere in his doctoral thesis in 1963 [23]. Nowadays, his central construct is usually called a Lawvere theory, more prosaically a single-sorted finite product theory [2,3]. It is a more flexible version of the universal algebraist's notion

[1] This work is supported by EPSRC grant GR/586372/01: A Theory of Effects for Programming Languages.
[2] Email: M.Hyland@dpmms.cam.ac.uk
[3] Email: ajp@inf.ed.ac.uk

1571-0661 © 2007 Elsevier B.V. Open access under CC BY-NC-ND license.
doi:10.1016/j.entcs.2007.02.019

of clone: indeed Lawvere himself arrived at the latter notion before formulating that which we describe below. In mathematical practice Lawvere theories arise whenever one has a functor into a category with finite products and one studies the natural transformations between finite products of the functor. (Historically the idea first arose in the form of cohomology operations.)

It is important to distinguish the invariant notion of Lawvere theory from the notion of equational theory. Equational theories are a form of presentation for Lawvere theories (or for clones): every equational theory determines a Lawvere theory and every Lawvere theory is determined by an infinite class of equational theories, that is, by those equational theories for which it is essentially the clone. Choosing good presentations for a (e.g. Lawvere) theory and deriving an invariant description of the theory from a presentation are important aspects of computer science which we shall discuss in passing; but the semantics of a theory can be considered independently of that.

The second category-theoretic formulation of universal algebra, which was in terms of monads, has a more complicated history. Monads typically arise from adjoint pairs of functors; and in such a case, the Eilenberg-Moore [8] and Kleisli [22] categories of algebras for the monad provide adjoint pairs which one can regard as approximations to the original adjoint pair. This notion of monad (or triple or standard construction) arose in algebraic topology for reasons distinct from universal algebra, see for instance [10]. In [8] Eilenberg and Moore noted that in case T is the free group monad, their category of T-algebras is the category of groups. Then Linton [29] made the general connection between monads and Lawvere theories (universal algebra): every Lawvere theory gives rise to a monad on Set whose category of algebras is equivalent to the category of models of the Lawvere theory, and, subject to a generalisation in the definition of Lawvere theory, every monad arises thus, uniquely up to coherent isomorphism.

Monads immediately became the more common category theoretic formulation of universal algebra, see for instance [31]. In retrospect, that surprises the current authors: the notion of Lawvere theory arose directly from universal algebra while that of monad did not; Lawvere theories relate more closely to universal algebra, and they immediately allow natural constructions that arise in universal algebra, such as those of taking the sum or tensor of theories. So the first main goal of the paper is to investigate how and why history favoured monads.

Moving forward to the late 1980's, computer scientists, led by Eugenio Moggi, began extensively to exploit the notion of monad but without reference to universal algebra [33,34,35]. Moggi wanted to unify the study of what he called notions of computation. The aim was to recover the many examples of denotational semantics already proposed at that time as instances of one general construction. He had in mind models for exceptions, side-effects, interactive input/output and non-determinism, as well as partiality and continuations. Probabalistic non-determinism [18] was soon to be added to the list. Monads have come to be used as a tool in computer science in contexts other than those originally proposed (data bases [5], pure function languages [48]), but it seems to us that their central use is Moggi's. We

maintain however that two of Moggi's applications, partiality and continuations are of a different nature from the others. Partiality arises from recursion without any imperative behaviour. We consider special features of continuations further in this paper.

In retrospect it seems to us that the universal algebra perspective is fundamental to the other notions of computation, those which we now prefer to call computational effects; but that was not clear at the time. The various computational effects and Moggi's corresponding monads arise from computationally natural operations, such as *raise* for exceptions, *lookup* and *update* for side-effects, *read* and *write* for interactive input/output, nondeterministic $\vee$ for nondeterminism, and $[0,1]$-many binary operations $+_r$ for probabilistic nondeterminism, subject to computationally natural equations. So there are evident scientific issues relating to the computational justification of (presentations of) Lawvere theories. Here we investigate how it was that Lawvere theories did not arise when Moggi proposed his approach, what did happen ten years later when they did arise, and how the relationship between computational effects and universal algebra might develop from here.

Gordon Plotkin was Moggi's PhD supervisor and later provided the computational expertise required to develop the study of computational effects in terms of universal algebra. The algebra of effects has been one of the major themes of his mature scientific research, hence the submission to this volume of our modest survey.

The paper is organised as follows. In Section 2, we recall the development of the notions of Lawvere theory and model. In Section 3, we explain properties and constructions on Lawvere theories that later proved useful in computation. In Section 4, we explain how each Lawvere theory gives rise to a monad on Set. In Section 5, we analyse how the notion of monad gained prominence over that of Lawvere theory in the category theoretic understanding of universal algebra. In Section 6, we investigate the use of monads, and later of Lawvere theories, the former by Moggi, the latter by Plotkin, in modelling computational effects. And in Section 7, we speculate upon the implications of the connection between computational effects and universal algebra.

We are grateful to Bill Lawvere, Mike Mislove and Eugenio Moggi for comments on early drafts of this paper. Their observations have materially affected our formulation of the salient issues.

2 Lawvere theories

The 1930's were a remarkable decade for foundational mathematics. At the time, mathematics was dominated by Germany to an extent that has never since been parallelled by any country. The era saw not only famous discoveries in logic, but also the development of two topics, fundamental from a category theoretic perspective, namely algebraic topology and universal algebra. The researchers involved with those developments were often the same or at least were closely related to each other. An interesting example is Saunders Mac Lane, who went to Germany to

write a thesis in logic, became one of the world's leading algebraic topologists, and co-wrote one of the world's most influential texts on algebra.

By the 1960's, algebraic topology and universal algebra had become much more distinct, and the new generation of researchers tended to have a deep understanding of one topic or the other, but not both. It was into that environment that Bill Lawvere entered. Lawvere was a student of Samuel Eilenberg at Columbia University in New York, at a time when Eilenberg was educating or influencing many of the foundational figures in North American category theory. Others who played a part in the development of the category theory we consider in this paper were Peter Freyd, recognised as an important predecesor in Lawvere's thesis, Michael Barr, Fred Linton, Jon Beck and Myles Tierney, the last two also students of Eilenberg. Almost all of them were expert in algebraic topology, but Lawvere's PhD thesis under Eilenberg was in universal algebra. (Late in the writing of this paper we learnt a piece of the history, which is particularly interesting from a computer science perspective. Eilenberg, who let it be known that he did not read Lawvere's thesis when it was written, did do so a few years later and for computer science reasons. Apparently this influenced his lectures 'Universal algebras and the theory of automata' at the AMS summer meeting in Toronto in 1967. As we write efforts are under way to locate and reassess this material.)

In his thesis, Lawvere axiomatised the clone of an equational theory along the following lines. Take a skeleton of the category of finite sets and all functions between them. So for each natural number n we have an object, n say, with n elements. The category has finite coproducts given on objects n, m, by cardinal sum $n + m$. Evidently it is equivalent to (any version of) the free category with finite coproducts on 1. We make a choice of coproduct structure: for definiteness we make the standard (ordinal sum) choice making $+$ strictly associative. (But nothing essential follows from that choice.)

Definition 2.1 Let $\aleph_0$ denote a skeleton of the category of finite sets and all functions between them, considered as a category with strictly associative coproducts.

Since $\aleph_0$ is equipped with finite coproducts, it is immediate that the opposite category $\aleph_0^{op}$ is equipped with finite products.

Definition 2.2 A *Lawvere theory* consists of a small category L with (necessarily strictly associative) finite products and a strict finite-product preserving identity-on-objects functor $I : \aleph_0^{op} \longrightarrow L$. A map of Lawvere theories from L to L' is a (necessarily strict) finite-product preserving functor from L to L' that commutes with the functors I and I'.

Thus the objects of any Lawvere theory L are exactly the objects of $\aleph_0$, and every function between such objects yields a map in L. One often refers to the maps of a Lawvere theory as *operations*, those arising from $\aleph_0$ being the *basic product operations*. The notion of map between Lawvere theories encapsulates the idea of a simple interpretation of one theory in another. Note that the behaviour of an interpretation on the product structure is determined.

Trivially, the definitions of Lawvere theory and map between them yield a category Law, with composition given by ordinary composition of functors. The functor I plays an important structural role in this regard: the category FP of small categories with finite products and strict finite-product preserving functors between them extends naturally to one of the leading examples of a 2-category [16], whereas the category Law, which is essentially a subcategory of FP, does not naturally extend in a compelling way. For the functor I fully determines the behaviour of finite products in L. It is immediate that given maps $F, G : L \longrightarrow L'$ between Lawvere theories, a coherent natural transformation from F to G (that is one which composes with $\aleph_0^{op} \longrightarrow L$ to give the identity) is completely determined by the identity on the object 1. So the only such are identities. This is a much more restrictive situation than in the usual categorical logic extensions to Lawvere theories (compare [2] and see our discussion in Section 5), and it is because of this that it makes sense to regard Law as a simple category. (Without coherence, natural transformations correspond to unary operations in L' which intertwine between the images of operations in L under F and G respectively; but there is little interest in these from the point of view of models.)

We stress the fact that in the definition of Lawvere theory, the functor I is not required to be an inclusion.

Example 2.3 There is a Lawvere theory $Triv$ that is equivalent to the unit category 1: its objects are the objects of $\aleph_0$, and there is one arrow from any object to any other object. The functor I is the identity-on-objects but is trivial on maps.

This is a useful example giving counter-examples to natural seeming conjectures. Although trivial, it is important to the structure of the category Law as it is the terminal object. The identity $\aleph_0^{op} \longrightarrow \aleph_0^{op}$ gives an initial object. More generally, Law enjoys good closure properties: it is complete and cocomplete, indeed it is a locally finitely presentable category. We shall investigate finite coproducts in Law briefly in Section 3.

In passing we note that there are just two Lawvere theories L such that the hom set $L(2, 1)$ has just one element. In addition to the theory $Triv$ one has the following variant.

Example 2.4 There is a Lawvere theory $Triv_0$ with no arrows from 0 to n for $n \neq 0$ and one arrow between objects in all other cases. The functor I is the identity-on-objects and identifies all maps with the same domain and codomain.

The relation between $Triv$ and $Triv_0$ is an example of a general phenomenon: in the language of [23], the second is the result of depleting the first by making inexpressible the definable constants. Leaving that aside we note that examples 2.3 and 2.4 are the only counterintuitive ones. It is generally harmless to pretend that the functor I in the definition of Lawvere theory is faithful, as it is so in all examples of primary interest.

For most mathematical purposes, one understands a Lawvere theory by study of its models.

Definition 2.5 A *model* of a Lawvere theory L in any category C with finite products is a finite-product preserving functor $M : L \longrightarrow C$.

Note that one asks for preservation of finite products here, not for strict preservation of them. That may seem surprising in the light of the strictness in the definition of Lawvere theory. One consequence is that a pair of models M and M' of a Lawvere theory L may differ only because of a choice of product in C. However, as Lawvere noted from the start [23], preservation rather than strict preservation of finite products is fundamental: if one demanded strict preservation, the category of models for the Lawvere theory for a monoid would be empty (!), rather than being the category of monoids as one wants. The reason is that, with the usual set-theoretic definitions, finite products in Set are not strictly associative, whereas they are strictly associative in any Lawvere theory. Preservation rather than strict preservation has other advantages: for example it allows a smooth account of change of base category along a finite preserving functor $H : C \longrightarrow C'$.

The requirement that M preserves projections, which is part of what preservation of products means, determines the behaviour of M on the basic product operation If for every function f: for projections in L amount to coprojections in $\aleph_0$, and every function f there is given by a family of coprojections. So what determines a model is the interpretation of the other operations.

The notion of Lawvere theory axiomatises the clone of an equational theory: that was its intent. Given any equational theory, one can generate a Lawvere theory: in the category L we construct, the object n is the n-fold *product* of n copies of 1 and so a map from m to n , corresponds uniquely to n maps from m to 1; and one of the latter amounts to an equivalence class of terms in (at most) m free variables generated by the operations subject to the equations of the theory.

Definition 2.6 For any Lawvere theory L and any category C with finite products, the category $Mod(L, C)$ is defined to have objects given by all models of L in C, with maps given by all natural transformations between them.

The correctness of the above definition of map in $Mod(L, C)$ is a more subtle matter than may at first appear. One can readily prove that the naturality condition implies that all natural transformations between models respect the finite product structure: for any natural transformation α between models M and N, and for any n in $\aleph_0$, the map $\alpha_n : Mn \longrightarrow Nn$ is given by the product of n copies of the map $\alpha_1 : M1 \longrightarrow N1$. Thus the maps in $Mod(L, C)$, which we defined to be all natural transformations, could equally be defined to be all natural transformations that respect the product structure of L. In the context of maps between models, we observe that a pair of models M and M' of L, differing (the issue mentioned above) only in a choice of product in C are isomorphic in $Mod(L, C)$.

The semantic category C of primary interest is Set. So consider a model M of a Lawvere theory L in Set. The set $M1$ determines Mn up to coherent isomorphism for every n in L: for M preserves finite products of L, equivalently of $\aleph_0^{op}$, these are finite coproducts of $\aleph_0$, which are given by finite sum, and so Mn must be the product of n copies of $M1$. Thus to give a model M is equivalent to giving a

set $X = M1$ together with, a function from X^m to X for each map of the form $f : m \longrightarrow 1$ in the category L, subject to the equations given by the composition and product structure of L; and $Mod(L, C)$ is equivalent to the evident category of such structures. This analysis routinely extends to any category C with finite products.

The flexibility as regards the category in which models may be sought is an important feature of the Lawvere theory approach. We recall the traditional example. Take the Lawvere theory L_G of a group, so that the category $Mod(L_G, Set)$ of models in Set is (equivalent to) the category of groups. (At the beginning of the next section we explain why the theory is determined by its category of models in $Sets$.) But we can interpret L_G in other categories. In particular a model of L_G in the category Top of topological spaces is essentially a topological group. We can tell the same story for other base categories, categories of sheaves, of differentiable manifolds, of schemes and so on. A special case of interest is that of models of L_G in the category $Group \simeq Mod(L_G, Set)$ of small groups. By the Eckmann-Hilton argument [6], which was published in 1962, these are abelian groups; and this explains why the higher homotopy groups are abelian. These examples give some indication how the notion of Lawvere theory brought precision and unity to constructs that were already being studied in the 1960s.

The model category $Mod(L, C)$ is functorial in both arguments, functoriality being induced by composition. An interpretation $L \longrightarrow L'$ induces a functor $Mod(L', C) \longrightarrow Mod(L, C)$, while a finite preserving functor $C \longrightarrow C'$ induces a functor $Mod(L, C) \longrightarrow Mod(L, C')$. We shall briefly contrast this with the situation for monads.

3 Properties of Lawvere theories

Now let us consider some of the properties of the notion of Lawvere theory, and some constructions that can be made with them. Unlike equational theories, Lawvere theories are semantically invariant. The precise sense in which that is so is as follows. With each Lawvere theory L, we associate the underlying set functor

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

given by evaluation at 1. (This association is the *semantics* functor of Lawvere [23].) We say that the categories $Mod(L, Set)$ and $Mod(L', Set)$ of models are coherently equivalent if they are so respecting the underlying set functor.

Proposition 3.1 *Given Lawvere theories L and L', if the categories $Mod(L, Set)$ and $Mod(L', Set)$ are coherently equivalent, then the Lawvere theories L and L' are isomorphic in the category Law.*

Proof. For any Lawvere theory L, the Yoneda embedding restricts to a fully faithful functor of the form

$$Y_L : L^{op} \longrightarrow Mod(L, Set) \, .$$

The representable $Y_L(n) = L(n, -)$ in $Mod(L, Set)$ itself represents the functor

$$ev_n \cong (ev_1)^n : Mod(L, Set) \longrightarrow Set\,.$$

So an equivalence between $Mod(L, Set)$ and $Mod(L', Set)$ respecting the underlying set functor ev_1, induces isomorphisms

$$Mod(L, Set)(L(n, -), L(m, -)) \cong Mod(L', Set)(L'(n, -), L'(m, -))$$

and so (by two uses of Yoneda) isomorphisms

$$L(n, m) \cong L'(n, m)$$

compatible with composition. The isomorphism respects the basic product structure because ev_1 is isomorphic to

$$Mod(I, Set) : Mod(L, Set) \longrightarrow Mod(\aleph_0^{op}, Set) \simeq Set\,.$$

□

The argument we have sketched is part of Lawvere's adjunction between semantics and algebraic structure [23]. The ideas are extended and refined in the treatment of algebraic operations in [38].

A major thrust of recent work (see [15] and [13]) has been to understand computationally natural ways to combine computational effects in terms of operations on Lawvere theories. We briefly review here two natural ways to combine Lawvere theories, which have proved of value.

First we explain a sum of Lawvere theories: it is is precisely the coproduct in the category Law. It has a simple description as it agrees with the obvious sum of equational theories: it corresponds to taking all operations from each of the equational theories at hand, subject to all equations of each of the theories, with no additional equations. So a model for $L + L'$ in Set is a set equipped with both the structure of a model of L and the structure of a model of L'.

One does require a little care with this notion of sum, but only the evident care one requires in regard to the sum of equational theories anyway. For instance, given Lawvere theories L and L', there are maps of Lawvere theories given by coprojections $L \longrightarrow L + L'$ and $L' \longrightarrow L + L'$. But those coprojection functors need not be faithful: if L is $Triv$, then $L + L'$ is also $Triv$, so the coprojection from L' is trivial. (There are analogous issues with $Triv_0$.)

One can also consider the tensor product $L \otimes L'$ of Lawvere theories L and L' which can be explained as follows, (but see also discussions in [9,45]). The category $\aleph_0$ not only has finite coproducts, but also has finite products, which we denote by $n \times n'$. The object $n \times n'$ may also be seen as the coproduct of n copies of n'. So, given an arbitrary map $f' : n' \longrightarrow m'$ in a Lawvere theory, it is immediately clear what we mean by the morphism $n \times f' : n \times n' \longrightarrow n \times m'$. We define $f \times n'$ by conjugation, and, in the following, we suppress the canonical isomorphisms.

Definition 3.2 Given Lawvere theories L and L', the Lawvere theory $L \otimes L'$, called the tensor product of L and 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 $f : n \longrightarrow m$ in L and $f' : n' \longrightarrow m'$ in L', we demand commutativity of the diagram

$$\begin{array}{ccc} n \times n' & \xrightarrow{n \times f'} & n \times m' \\ {\scriptstyle f \times n'}\downarrow & & \downarrow{\scriptstyle f \times m'} \\ m \times n' & \xrightarrow[m \times f']{} & m \times m' \end{array}$$

The tensor product always exists either because it is defined by operations and equations, or more elegantly by appeal to the work on pseudo-commutativity in [16].

Proposition 3.3 *The tensor product $\otimes$ extends canonically to a symmetric monoidal structure on the category of Lawvere theories.*

A proof for this proposition is elementary. The unit for the tensor product is the initial Lawvere theory, that is, the theory generated by no operations and no equations. We can take this to be $\aleph_0^{op} \longrightarrow \aleph_0^{op}$, so it is the initial object of the category of Lawvere theories; and so it is also the unit for the sum.

This last result gives some indication of the definitiveness of the tensor product, but does not amount to much. What is central to the meaning of commutativity but is not a common feature of operations on theories, is a simple characterisation of $L \otimes L'$ in terms of the categories of models of L and L' [14,15].

Theorem 3.4 *For any category C with finite products, there is a coherent equivalence of categories between $Mod(L \otimes L', C)$ and $Mod(L, Mod(L', C))$.*

This theorem gives us a new view on the Eckmann-Hilton argument mentioned at the end of Section 2. What that argument shows is that the tensor product of the Lawvere theory L_G of a group with itself is isomorphic to the Lawvere theory L_A of an Abelian group.

As a final general remark we observe that properties of and structure on the category of Lawvere theories remains an interesting area of study in its own right. A survey together with a list of problems appears in [25], and the commentary to the TAC reprint updates the material.

4 Monads

Soon after Lawvere gave his characterisation of the clone of an algebraic theory, Linton showed that every Lawvere theory yields a monad on Set [29]. The construction extends to a fully faithful functor from Law to the category Mnd of monads on Set.

This functor is not an equivalence of categories. So in this precise sense, a monad on Set can be regarded as a more general notion than that of Lawvere theory.

Linton also gave a partial converse. One can generalise the definition of Lawvere theory to allow for arities of arbitrary size, with a generalised theory no longer a small category or fully determined by one. The construction of a monad from a Lawvere theory then generalises to an equivalence of categories. In [30], Linton gives generalisations of Lawvere's treatments of semantics and algebraic structure. It is implied that the case treated by Lawvere should be seen as a special case of the more general theory.

We give some details of the relation between theories and monads as analysed by Linton. Given any Lawvere theory L, there is a canonical forgetful functor $U_L : Mod(L, C) \longrightarrow C$ given by evaluation at the object 1 of L, equally of $\aleph_0$. If that forgetful functor has a left adjoint F_L, as it does whenever C is locally presentable, one can prove either directly or by Beck's monadicity theorem [2] that it exhibits $Mod(L, C)$ as equivalent to the category T_L-Alg for the induced monad T_L on C. Since in particular Set is locally finitely presentable, every Lawvere theory L induces a monad T_L on Set.

Proposition 4.1 *The monad T_L may be described by the following colimit (coend):*

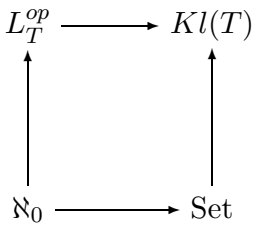
$$T_L X = \int^{n \epsilon \aleph_0} L(n, 1) \times X^n$$

This colimit is the coproduct over all natural numbers n of the set $L(n, 1) \times X^n$, factored by identifying elements determined by taking projections and diagonal maps of $\aleph_0^{op}$. It is easy to see that this corresponds to the collection of all terms in the theory up to equality in the theory. Monads and pseudo-monads which arise from many kinds of generalised algebra are constructed using analogous formulae, but we do not pursue that here. We note that the formula allows an easy proof that the construction of a monad from a Lawvere theory is functorial and that it is fully faithful as a functor from Law to Mnd; and with a little more effort one can prove the following.

Proposition 4.2 *The construction sending a Lawvere theory L to the monad T_L determined by the forgetful functor $U_L : Mod(L, Set) \longrightarrow Set$ extends to a fully faithful functor from Law to Mnd. Moreover, the comparison functor exhibits an equivalence between $Mod(L, Set)$ and T_L-Alg.*

One can readily check that the monad T_L is finitary for every Lawvere theory L. When the base category is Set, finitariness characterises the image of the construction, but that was an observation of a later time [19].

For a converse, first observe that for any monad T on Set, the Kleisli category $Kl(T)$ has all small coproducts and the canonical functor $I : Set \longrightarrow Kl(T)$ preserves them: for the canonical functor I has a right adjoint and is identity-on-objects. Then restricting I to the full subcategory $\aleph_0$, we obtain (the opposite of) a Lawvere theory as in the diagram.



It is then straightforward to show the following.

Proposition 4.3 *The construction sending a monad T on Set to the category $Kl(T)^{op}_{\aleph_0}$ determined by restricting $Kl(T)$ to the objects of $\aleph_0$ extends to a functor from Mnd to Law.*

The simple idea behind the construction is this. If T_L is the monad for a Lawvere theory L, then $Kl(T)(n,m)$ consists of maps between the free models for L on n and m; and as $Kl(T)(n,m) \cong Set(n, Tm)$, these correspond to n-tuples of elements of Tm. But the free model is given by equivalence classes of terms so this is just $L(m,n)$. Thus it is routine to verify that, for any Lawvere theory L, the Lawvere theory L_{T_L} is isomorphic in Law to L. However the corresponding statement starting with a monad T on Set is not true because the construction T_L yields precisely finitary monads. Indeed we have the following.

Theorem 4.4 *The constructions sending L to T_L and that sending T to L_T exhibit Law as a full coreflective subcategory of Mnd, the category of monads on Set.*

The point of view extending this taken by Linton was as follows. One can define a generalised theory to consist of a locally small category L with all small products, together with a strict product preserving identity-on-objects functor from the opposite of a skeleton of Set to L. With this notion of generalised Lawvere theory, one can show that the construction of Proposition 4.2 extends [13]; and then in the corresponding version of Theorem 4.4 we have an equivalence of categories [29].

This extension seems smooth and attractive, so we briefly give some counter-indications. The problems do not occur for monads of bounded rank (monads with rank); for them a version of the material for (finitary) Lawvere theories goes through. However there are interesting monads without rank [2,13]. In mathematics one has the monad for compact Hausdorff spaces, and in the theory of computation, those for continuations. One meets the monad for sup lattices in both areas of science. The problem as we see it is that one cannot handle the notion of large Lawvere theory corresponding to the general notion of monad as one can in the case where the monad or theory has rank. So this move inherently involves a step away from the idea of universal algebra.

One point is clear. The notion of large Lawvere theory is not closed under sum. Indeed there is no sum of the monad generated by a unary operation with any of the monads without rank mentioned above. This is explained for continuations in [13], but the argument in the other cases is essentially the same. Rather than repeat the proof here we give an intuition why it is the case, based on the analysis of the sum

given in [15]. In the large Lawvere theory corresponding to a monad T without rank, we have a class of operations; and this gives a monad on Set only because there are equations between complex expressions in this class of operations. Now we add an independent unary operation, s say. Consider the class of operations obtained from the original class by postcomposing with s. Since s is independent there are now no non-trivial equations between complex expressions in these new operations. As a result we lose control over the size of what may be constructed in the putative sum theory. Specifically, the unit of T without rank will be monic and so it follows that starting from any set x the free algebra on x for the sum must contain $T^*x = \mu y.(Ty + x)$ the image of x under the free monad T^* generated by the functor T. But $T^*x \cong T(T^*x + x)$ and this is impossible for cardinality reasons in all the cases we know of. Thus none of these sums exist.

An intuition of a similar kind makes it clear that if one of the generalised Lawvere theories L and L' has rank then the tensor product $L \otimes L'$ exists. (This is proved formally in [13].) However the intuition does not carry through if both L and L' are without rank, and we believe that the tensor product does not exist in general.

The different range of generality of the ideas of monads and Lawvere theories bears on the connection between (generalised) Lawvere theories and monads. One can consider monads on any category, while Lawvere theories correspond to (finitary) monads on Set. On the other hand, a monad on a category has algebras (i.e. models) just in that category, while a Lawvere theory naturally has models in any category with products. So while monad maps between monads on Set (see [2] for this notion of monad map) correspond directly to maps of Lawvere theories, there is nothing in the world of monads (at least nothing to which one has immediate access) corresponding to the functoriality of $Mod(L, C)$ in the category with products C.

We make some remarks by way of amplification of this last point. If $F : C \to D$ is product preserving between (say) locally presentable categories, then it follows from the description in Proposition 4.1 that $Mod(L, F) : Mod(L, C) \to Mod(L, D)$ is a lift of F, and so corresponds to a monad map (in the general sense, see [46]) between the induced monads T_L^C on C and T_L^D on D. Now there appears to be no way to get a handle even on this purely in terms of monads: the description of the monad (and indeed the monad map) depends on the theory. But the use of Lawvere theories takes us much further. By way of illustration, we consider some issues for the category Top of topological spaces.

The (discrete) embedding $Set \to Top$ of Set in Top preserves finite products and so we get an obvious embedding of the category of groups (in Set) into that of topological groups (which as we mentioned earlier is the models in Top for the relevant Lawvere theory). Now Top is not locally presentable, so in the absence of general theory we need to check that the coend formula of Proposition 4.1 enables us to extend the monad for groups from Set to Top. That is reasonably straightforward, but $Set \to Top$ does not preserve other products. So if we consider a countable Lawvere theory (which we need when treating side-effects, see Example 6.2), we cannot expect to lift the embedding at the level of models. However the coend

formula still applies and we still get a monad on Top. But one should take care. Since the forgetful $Top \to Set$ preserves all products, the space has the expected points. But the coend does not generally give the expected topology: for example from the side-effects monad on Set we do not get the natural side-effects monad on Top. For a yet more teasing situation the reader may like to consider what happens with the continuations monad.

5 Lawvere theories and monads

By the mid 1960s the categorical understanding of universal algebra was established. (Most of the relevant publications appeared in 1966.) Lawvere theories axiomatised the notion of a clone of an equational theory. Monads, which had arisen in algebraic topology, had been seen to generalise the notion of Lawvere theory; and, as we indicated in Section 4, it had been recognised that one could formally extend the notion of Lawvere theory to give a notion of theory equivalent to that of monad on Set. But in retrospect, the generalised notion of theory involved in these developments is not attractive. It is a further step away from the activity of universal algebra, which is itself an abstraction from the bulk of activity within algebra. And one loses useful constructions: a sum of monads need not exist in general, and it seems likely that even the tensor need not exist either. That is worrying partly because these are natural algebraic constructs, and partly because even when they do exist, it is not easy to see how and why one might naturally think of their universal properties in terms of monads. Notwithstanding all this however, category theorists overwhelmingly began to conceptualise universal algebra in terms of monads rather than Lawvere theories, see for instance [31] where the notion is mentioned in one sentence. So why was that? It is always difficult to understand the motivations of people either individually or collectively, but there were salient scientific historical facts in the particular circumstances of the time.

First, even had the will been there to conceptualise universal algebra in terms of Lawvere theories, the ethos of category theory is to see constructs in terms of arbitrary categories with axiomatically defined structure, then to take Set as a leading example. That is easy for the notion of monad but difficult for the notion of Lawvere theory, as it requires one to answer the question:

> Given an arbitrary object X of an arbitrary category C satisfying axiomatic conditions, what does it mean for X to be finite?

A definitive answer to that question only appeared in 1971, when Gabriel and Ulmer published, in German, their account of locally finitely presentable categories [11].

Second, enriched categories had not been appreciated and developed. The notion of enriched category was first formulated by Eilenberg and Kelly in 1966 in [7], but it took time for it to enter the mainstream, and a definitive account of finiteness in the enriched setting only appeared in 1982 in [20]. The reason enriched categories help is because, given a notion of finiteness, the generalisation of Lawvere theory to that setting is easy and covers a wide class of the examples of structures that are usefully treated under the rubric of universal algebra. The notion of enriched

Lawvere theory was only published in 2000 in [43], and while it was understood well before that, the understanding is still not widely spread. We shall give some details of enriched Lawvere theories shortly.

Third, developments in categorical logic which favour the Lawvere theory perspective were yet to come. As discussed in the Appendix to [28] there are two extant senses to logic, a broader one concerning the development of thinking and a narrower one typically concerned with the logic of properties. In the epoch we consider, the former was undergoing substantial development largely stimulated by Lawvere (see [24,25,26,27]). But it was only in the 1970s that a sophisticated categorical logic in the narrower sense emerged; and ironically Lawvere theories fit naturally within this narrow reading of logic. Lawvere theories correspond to single-sorted equational theories; more generally finite limit theories [11] mildly generalise Horn clauses; and more generally still, one has regular theories, coherent theories, and geometric theories, all fitting into the topos theory perspective on categorical logic [32,2]. In the context of that work, monads are isolated. Though they are part of logic in the broader sense, in contrast to Lawvere theories, they do not immediately fit into this logical hierarchy. But that development all came later, so did not provide a mathematical culture which favoured Lawvere theories as opposed to monads in the mid 1960s.

Our first two reasons hang together, and we say a little about the technical developments on those fronts as they are important for applications. Enriched Lawvere theories are defined as follows. One generalises from Set to a category V that is locally finitely presentable as a (symmetric) monoidal closed category [20]. That includes categories such as Cat, $Poset$, $Graph$, as well as all categories of algebras for any ordinary commutative equational signature. The notion of *cotensor* is then the definitive enrichment of the notion of a power.

Definition 5.1 Given an object X of V and an object B of a V-category C, the X-cotensor of B, if it exists, is the object B^X of C together with a family of isomorphisms

$$C(A, B)^X \cong C(A, B^X)$$

V-natural in A.

Taking V to be Set, the cotensor just defines the X-fold product of copies of B. Next an X-cotensor is called *finite* if X is finitely presentable, in the definitive sense of Gabriel and Ulmer. Now the basic material above generalises.

Definition 5.2 A *Lawvere-V-theory* is a small V-category L with finite cotensors together with a strict finite-cotensor preserving and identity-on-objects V-functor $I : V_f^{op} \longrightarrow L$, where V_f is a skeleton of the full sub-V-category of V determined by the finitely presentable objects of V.

Definition 5.3 A *model* of L in a V-category C with finite cotensors is a finite-cotensor preserving V-functor $M : L \longrightarrow C$.

One can duly define a V-category $Mod(L, C)$ and one can then generalise all our category theoretic analysis of ordinary Lawvere theories without fuss. The one

important aspect of ordinary Lawvere theories that has not yet been extended well is the syntactic notion of equational theory: as best we know, there is currently no enriched notion of equational theory corresponding to enriched Lawvere theories. But equational theories can be rephrased in terms of sketches, for which an enriched account does exist or at least can readily be gleaned from the literature [21]. This does appear to be a satisfactory approach to the syntactic treatment of enriched Lawvere theories.

In the presence of both enrichment and an axiomatic account of finiteness, one can proceed easily, but as we said that combination did not exist in 1966. In passing we note that if one has finiteness but not enrichment, one can still give an account of Lawvere theories relative to an arbitrary base category subject to axiomatically defined conditions. But that is awkward: it was only finally resolved in 2005, the details appearing in [36].

How history might have developed had finiteness and enrichment been resolved by 1966 and had the motivating experience of more general categorical logic been available then is imponderable. Lawvere theories might or might not have emerged as the definitive category theoretic formulation of universal algebra; but they would certainly have had a better chance. In the event, Lawvere did not proceed with universal algebra (though he did take an interest in enrichment), but rather went on to characterise the category *Set* [24], a line of work which eventually led to the notion of elementary topos and then to categorical logic. The category theoretic development of universal algebra was taken forward by Beck, Barr, and others whose expertise lay primarily in algebraic topology, where the notion of monad had arisen; that led to important mathematical developments, see for instance [1,4] and the account in [2], but the focus of these results is quite different from that of the Lawvere theory notion.

6 Computational Effects

We now jump from the mid 1960s to the late 1980s. In the meantime, the notion of monad had been consolidated as the primary category-theoretic formulation of universal algebra, as witnessed by its prominent role in Mac Lane's influential text [31], where Lawvere theories are only mentioned in the final line of the chapter on monads. Simultaneously and independently, computer scientists had begun to take an interest in category theory in connection with denotational semantics. Then, in 1987, Eugenio Moggi completed his PhD thesis at the University of Edinburgh under Gordon Plotkin, with Martin Hyland his external examiner. At precisely that point, Moggi's new idea of computational effects came to the attention of experienced category theorists.

A defining moment came at Moggi's oral defence. Moggi had completed a technical thesis on partiality, and the discussion turned to future work. He then introduced his new idea of notions of computation and proposed using monads to model them. It immediately struck Hyland as a particularly elegant idea, involving an enrichment of a basic type theory with terms having computational meaning. He

was very encouraging. It did not occur to Hyland, in the circumstances of the time, to mention universal algebra or Lawvere theories: rather he tried to learn about the computational concepts from the monads, having no independent understanding of them on the basis of which to apply critical judgement.

A year later, at the *Constructive Logic and Category Theory* workshop at the Isle of Thorns, Peter Freyd, who was the leading North American category theorist connected with computer science at the time, first heard Moggi's idea. He too was impressed and said so to Hyland who was sitting by him, and to others. But again the ideas of universal algebra and Lawvere theories did not crop up in the discussions of Moggi's ideas.

Moggi's work was soon published prominently in [33,35] and has been enormously influential as a meta-principle for thinking about programming languages. And it was some ten years before the move from monads to the Lawvere theories that give rise to them began to change the emphasis of this line of research. It is only now after a further five years of research that we begin to see the impact of that change. From Moggi's assignment to each set X of the set TX of values associated with a computational effect, one has passed to the study of the operations associated with the computational effect. Given that in most cases a choice of TX is apparent on the basis of some kind of computational intuition, the process is a kind of reverse engineering. Moreover one does not merely want the clone of an algebraic theory but rather a good choice of primitives in which to couch the theory, with a good choice of axioms. The recognition (see section 7) that Moggi's monad for side-effects arises from the computationally natural operations *lookup* and *update* subject to computationally natural equations was critical to the development of the ideas: we describe this case in Example 6.2 below. The general situation of knowing the monad and reconstructing the theory is not that familiar in mathematics: theory has had to be developed adequately to ground the practice, and the computationally significant examples have had to be worked through. This has resulted in a series of recent papers [37,38,39,40,41,42]. But one should recognise that issues of presentation are also mathematically significant. Famously for instance the theory of groups has a presentation in terms of division, but that presentation generally seems to be of little interest.

We now survey examples of computational effects, given by Moggi in terms of monads, and later described in terms of (countable) Lawvere theories. (For more extended discussions see [14,15].)

Example 6.1 The Lawvere theory L_E for exceptions is the free Lawvere theory generated by E operations $raise : 0 \longrightarrow 1$, where E is a set of *exceptions*. In terms of operations and equations, this corresponds to an E-indexed family of nullary operations with no equations. The monad on Set generated by L_E is $T_E = - + E$. More generally, the forgetful functor $U_L : Mod(L_E, C) \longrightarrow C$ induces the monad $- + \underline{E}$ on C, where $\underline{E}$ is the E-fold copower of 1, i.e., $\coprod_E 1$, assuming the coproduct exists in C.

For side-effects, one must make the routine generalisation from Lawvere theories

to countable Lawvere theories: the latter allow operations of countable arity [14,15]. In the following Val is a countable set of values and Loc is a finite set of locations. We make the harmless identification of Val with $\aleph_0$ and Loc with its cardinality.

Example 6.2 The countable Lawvere theory L_S for side-effects, where the state $S = Val^{Loc}$, is the free countable Lawvere theory generated by the operations $lookup : Val \longrightarrow Loc$ and $update : 1 \longrightarrow Loc \times Val$ subject to the seven natural equations listed in [39], four of them specifying interaction equations for *lookup* and *update* and three of them specifying commutation equations. It is shown in [39] that this countable Lawvere theory induces Moggi's side-effects monad $(S \times -)^S$ on Set. More generally, if C is any category with countable powers and copowers then, slightly generalising the result in [39], the forgetful functor $U_L : Mod(L_S, C) \longrightarrow C$ induces the monad $(S \times -)^S$ on C, where we write $(S \times -)$ for the S-fold copower $\coprod_S -$, and $(-)^S$ for the S-fold power $\prod_S -$.

In the previous example we assumed that the set of locations was finite. In his contribution to the Plotkin Symposium, Eugenio Moggi drew attention to a more subtle side-effects monad, which arises in case the set Loc of locations is infinite.

Example 6.3 Suppose we take Loc to be infinite. Then the operations and equations of [39] induce a submonad of the side-effects monad $(S \times -)^S$. The functor part is a finitary version of $(S \times -)^S$ given by pairs $(\sigma : S \to S, \alpha : S \to A)$ with the property that for each s,

(i) $\sigma(s)$ differs from s at a finite number of locations,

(ii) there is a finite number of locations such that if we fix s at these then
  (a) that fixes $\alpha(s)$, and $\sigma(s)$ on a finite set of locations as in (i), and
  (b) moreover the remaining locations are untouched.

The monad structure is inherited from $(S \times -)^S$.

The last two examples illustrate the give and take between the semantic analysis of a monad with a view to extracting an attractive presentation of the corresponding theory and the construction of a monad from a theory. In the first place one would hardly arrive at the second monad independently of the theory for side-effects. Secondly the language for the theory is not at all evident from the monad. Indeed since $S = Val^{Loc}$ is countable whenever Loc is finite non-empty, we get the same Lawvere theory in all non-degenerate cases of the basic side effects monad: the chosen presentation depends on the number of locations. (That is not the end of the semantic issues as regards state. If one thinks of the collection of locations as being potentially rather than actually infinite, one might argue for a presheaf approach as in [39] in place of the modified side-effects monad.)

For the next example, given any endofunctor F on a category C, let $\mu y.Fy$ denote the initial F-algebra if it exists. Then, for an endofunctor Σ on a category C with binary sums, the free Σ-algebra on an object x is $\mu y.(\Sigma y + x)$, with one existing if and only the other does. These free algebras certainly exist if C is locally countably presentable and Σ has countable rank, and in analogous cases.

Example 6.4 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 Lawvere theory: $T_{I/O}(X)$ is the free Σ-algebra on X, where $\Sigma Y = O \times Y + Y^I$ is the *signature* functor determined by the two operations; an algebra for Σ consists of an O-indexed family of unary operations and an I-ary operation. This is also the form of $T_{I/O}$ in the more general situation where it arises from $Mod(L_S, C)$ for a locally countably presentable category C.

Example 6.5 The countable Lawvere theory L_N for (binary) nondeterminism is the countable Lawvere theory freely generated by a binary operation $\vee : 2 \longrightarrow 1$ subject to equations for associativity, commutativity and idempotence, i.e., the countable Lawvere theory for a semilattice; the induced monad on Set is the finite non-empty subset monad $\mathcal{F}^+$.

Example 6.6 The countable Lawvere theory L_P for probabilistic nondeterminism is that freely generated by $[0,1]$-many binary operations $+_r : 2 \longrightarrow 1$ subject to the equations for forms of associativity, commutativity and idempotence as in [12]. The induced monad on Set is the monad $\mathcal{D}_f$ of distributions with finite support. For the record, the associative laws are $(a +_r b) +_s c = a +_{rs} (b +_t c)$, where $(1-r)s = (1-rs)t$; the pseudo-commutative laws are $a +_r b = b +_{(1-r)} a$, and idempotence is the family of equations $a +_r a = a$.

The category Set is not the category of primary interest in denotational semantics. One is more interested in ωCpo, and variants, which model recursion and non-termination. As we observed earlier, the relationship between finite Lawvere theories and finitary monads generalises without fuss to one between countable Lawvere theories and monads and then to one between countable enriched Lawvere theories and enriched monads, equivalently strong monads, on the category in which the enrichment takes place. For that theory to work, it suffices that the category be locally countably presentable as a cartesian closed category. The category ωCpo is an example of such a category. So the work here generalises to include ωCpo.

One thing which tells in favour of the Lawvere theory point of view is that the natural combinations of computational effects correspond to natural combinations of Lawvere theories, notably sum and tensor [14,15]. In particular, the combination of exceptions with any other computational effect corresponds to the sum of Lawvere theories; that is also true for the natural combination of interactive input/output with most other effects. On the other hand for combining side-effects, we have the following [15].

Example 6.7 Let L_S denote the countable Lawvere theory for side-effects, where $S = Val^{Loc}$, and let L denote any countable Lawvere theory. Then the monad $T_{L_S \otimes L}$ is isomorphic to $(T_L(S \times -))^S$.

This shows that the tensor product of computational effects agrees with Moggi's definition of the side-effects monad transformer. Furthermore we have the following

agreeable corollary.

Corollary 6.8 *The side-effects theory for $S = Val^{Loc}$ is the Loc-fold tensor product of the side-effects theory for $S = Val$.*

Returning briefly to the sum of theories we note that in the non-degenerate cases of interest it provides a way of combining theories without adding new equations to either. There is a way to do that (again in non-degenerate cases) purely in terms of monads by using a distributive law. From a distributive law $TS \to ST$ one gets a composite monad ST; but from the point of view of Lawvere theories this distributes the operations of L_S over those of L_T, and so the result is very different from the monad corresponding to the sum $L_S + L_T$. Moreover in important instances distributive laws do not exist. It was shown by Plotkin and Varacca that there is no distributive law for any of the nondeterministic choice monads over the probabilistic choice monad or vice versa. One needs considerable effort to resolve the situation [47]. In contrast with the situation for distributive laws, the various kinds of composite of Lawvere theories, such as sum, tensor, and a distributive tensor [17], always exist. Those we know include the combinations of computational effects of primary interest; and it seems likely that further combinations of computational effects which may well arise in particular in the enriched setting will also be best treated in terms of Lawvere theories.

One monad missing from the list of examples above is the continuations monad. This is a monad without rank, and so cannot be taken to be a useful kind of Lawvere theory. Indeed by the discussion in [13], it appears that the continuations monad transformer should be seen as something sui generis. It does not appear to be the result of combining the continuations monad with others, at least not in any very obvious sense. The change in emphasis, to which we referred above, leads one to differentiate between monads arising from operations and equations and the others, with the others including continuations. Furthermore, the resulting point of view downplays the role of the Kleisli construction for computational effects: generally one need not pass from a Lawvere theory to a monad, then to its Kleisli construction, but rather just start with the Lawvere theory L, knowing that it is the finitary restriction of $KL(T_L)^{op}$, then extending that directly if needed [44].

7 The future

In view of recent developments, it is tempting to wonder what might have happened in 1987 at and immediately after Moggi's oral defence had Mac Lane's book had a chapter on Lawvere theories with a section on monads rather than having a chapter on monads with a line about Lawvere theories? Would research have taken a different course had Hyland or Freyd, upon listening to Moggi, immediately asked about universal algebra or Lawvere theories? Of course we can never know, but we can say what did happen in 2001 when a universal algebra connection was eventually suggested.

In 2001, Gordon Plotkin gave the opening, invited talk at ETAPS in Genova,

with Moggi in the audience. Plotkin presented joint work with Power that began to relate computational effects and universal algebra [37]: he described structural operational semantics at the level of generality of universal algebra and studied examples drawn from the various forms of nondeterminism and probabilistic nondeterminism. At the end of his talk, Moggi suggested that his side-effects monad might be generated by the operations *lookup* and *update* subject to computationally natural operations. It was over the following few months that Plotkin and Power verified that and, later joined by Hyland, saw that what were then regarded as computational effects could, with one exception, fruitfully be seen as an instance and a development of universal algebra: interactive input/output was soon recognised as an example involving no equations; it was seen how to incorporate local state naturally; and the various ways of combining computational effects proved to be simple instances of combining Lawvere theories. The exception was continuations. So one wonders:

> What might have happened in 1987 had the connection with universal algebra been noticed at the time? More to the point, one wonders what will happen from now?

We end the paper with speculative answers to those questions:

- Perhaps computational effects might be seen as an instance and development of universal algebra? Continuations would not be regarded as a computational effect but rather as a distinct notion. It would still have its own body of theory, and one would still study the relationship between it and computational effects; but perhaps it would not be regarded *as* a computational effect?
- Monads appear quite directly in the study of continuations. So perhaps the notion of monad might be seen as a generalised semantics of continuations? With the unit and counit being regarded as generalised forms of thunk and force?
- Perhaps the construction sending a Lawvere theory L to the monad T_L might be seen as providing mathematical support to a CPS-transform? One could give semantics for the λ_c-calculus and the computational meta-language that look quite different to each other, the former in terms of closed $Freyd$-categories [42,44], and the latter in terms of monads. One might focus on the first-order fragment of the λ_c-calculus: Lawvere theories immediately provide models, and one can readily produce a sound and complete class of them [44].

References

[1] Barr, M., and J. Beck, *Acyclic models and triples*, Proc. Conf. on Categorical Algebra at La Jolla (1966), 336–344.

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

[3] Barr, M., and C. Wells, "Category Theory for Computing Science," Prentice-Hall, 1990.

[4] Beck, J., "Triples, Algebra and Cohomology," PhD thesis, Columbia University, New York, 1967.

[5] Bunemann, P., S. Naqvi, V. Tannen and L. Wong, *Principles of programming with complex objects and collection types*, Theoretical Computer Science **149** (1995), 3-48.

[6] Eckmann, B., and P. Hilton, *Group-like structures in general categories i: multiplications and comultiplications*, Math. Annalen **145** (1962), 227–255.

[7] Eilenberg, S., and G. M. Kelly, *Closed categories*, in Proc. Conf. on Categorical Algebra at La Jolla (1966), 421–562.

[8] Eilenberg, S., and J. C. Moore, *Adjoint Functors and Triples*, Illinois J. Math. **9** (1965), 381–398.

[9] Freyd, P. J., *Algebra-valued functors in general and tensor products in particular*, Colloq. Math. Wroclaw **14** (1966), 89–106.

[10] Godement, R., "Théorie des faisceaux," Herman, Paris, 1958.

[11] Gabriel, P., and F. Ulmer, "Lokal präsentierbare Kategorien," Lecture Notes in Mathematics **221**, Springer-Verlag, Berlin, 1971.

[12] Heckmann, R., *Probabilistic Domains*, Proc. CAAP '94, Lecture Notes in Computer Science **136** (1994), 21-56.

[13] Hyland, M., P. B. Levy, G. D. Plotkin and A. J. Power, *Combining algebraic effects with continuations*, Accepted in Theoretical Computer Science.

[14] Hyland, M., G. D. Plotkin and A. J. Power, *Combining computational effects: commutativity and sum*, Proc. 2nd IFIP Conf on Theoretical Computer Science, (eds. Ricardo A. Baeza-Yates, Ugo Montanari and Nicola Santoro), Kluwer (2002), 474–484.

[15] Hyland, M., G. D. Plotkin and A. J. Power, *Combining effects: sum and tensor*, Theoretical Computer Science, **357** (2006), 70-99.

[16] Hyland, M., and A. J. Power, *Pseudo-closed 2-categories and pseudo-commutativities*, J. Pure Appl. Algebra **175** (2002), 141–185.

[17] Hyland, M., and A. J. Power, *Discrete Lawvere theories and computational effects*, Theoretical Computer Science, **366** (2006), 144–162.

[18] Jones, C., and G. D. Plotkin, *A probabalistic power domain of evaluations*. Proceedings of the 4th Symposium on Logic in Computer Science, IEEE Publications 1989, 186-195.

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

[20] Kelly, G. M., *Structures defined by finite limits in the enriched context I*, Cahiers Topologie Geom. Differentielle **23** (1982), 3–41.

[21] Kinoshita, Y., A. J. Power and M. Takeyama, *Sketches*, J. Pure Appl. Algebra **143** (1999), 275–291.

[22] Kleisli H., *Every standard construction is induced by a pair of adjoint functors*, Proc. Amer. Math. Soc. **16** (1965), 544-546.

[23] Lawvere, F. W., "Functorial Semantics of Algebraic Theories," PhD Thesis, Columbia University, 1963. (Available with commentary as TAC Reprint 5.)

[24] Lawvere, F. W., *An elementary theory of the category of sets*, Proc. Nat. Acad. Sci. USA **50** (1964), 1506–1511. (Long version with commentary available as TAC Reprint 11.)

[25] Lawvere, F. W., *Some algebraic problems in the context of functorial semantics of algebraic structures*, Reports of the Midwest Category Seminar II, (ed. S. Mac Lane), Lecture Notes in Mathematics **61**, Springer-Verlag 1968, 41-61. (Available with commentary as addition to [23] in TAC Reprint 5.)

[26] Lawvere, F. W., *Adjointness in Foundations*, Dialectica **23**, 1969, 281-296. (Available with commentary as TAC Reprint 16.)

[27] Lawvere, F. W., *Equality in hyperdoctrines and the comprehension schema as an adjoint functor*, Applications of Categorical Algebra, (ed. A. Heller), Proceedings of Symposia in Pure Mathematics **17**, American Mathematical Society 1970, 1-14.

[28] Lawvere, F. W., and R. Roseburgh, "Sets for Mathematics," Cambridge University Press, 2003.

[29] Linton, F. E. J., *Some aspects of equational theories*, Proc. Conf. on Categorical Algebra at La Jolla (1966), 84–95.

[30] Linton, F. E. J., *An outline of functorial semantics*, Seminar on Triples and Categorical Homology Theory, Lecture Notes in Mathematics **80**, 7-52.

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

[32] Makkai, M., and G. Reyes, "First-Order Categorical Logic," Lecture Notes in Mathematics **611**, Springer-Verlag, Berlin, 1977.

[33] Moggi, E., *Computational lambda-calculus and monads*, Proc. LICS '89 (1989), 14–23.

[34] Moggi, E., *An abstract view of programming languages*, University of Edinburgh, Report ECS-LFCS-90-113, 1989.

[35] Moggi, E., *Notions of computation and monads*, Inf. and Comp. **93** (1991), 55–92.

[36] Nishizawa, K., and A. J. Power, *Lawvere theories over a general base*, J. Pure Appl. Algebra, to appear.

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

[38] Plotkin, G. D., and A. J. Power, *Semantics for Algebraic Operations*, Proc. MFPS XVII, Electronic Notes in Theoretical Computer Science **45** (2001) 1–14.

[39] Plotkin, G. D., and A. J. Power, *Notions of computation determine monads*, Proc. FOSSACS 2002, Lecture Notes in Computer Science **2303** (2002), 342–356.

[40] Plotkin, G. D., and A. J. Power, *Computational effects and operations: an overview*, Proc. Workshop Domains 2002, Electronic Notes in Theoretical Computer Science **73** (2002), 149–163.

[41] Plotkin, G. D., and A. J. Power, *Algebraic operations and generic effects*, in *Proc. MFCSIT 2000*, Applied Categorical Structures **11** (2003) 69–94.

[42] Plotkin, G. D., and A. J. Power, *Logic for computational effects: work in progress*, Proc. IWFM 2003, EWiC (2003).

[43] Power, A. J., *Enriched Lawvere Theories*, Theory and Applications of Categories **6** (1999), 83–93.

[44] Power, A. J., *Generic models for Computational Effects*, Theoretical Computer Science **364** (2006), 254-269.

[45] Schubert, H., "Categories," Springer-Verlag, Berlin, 1972.

[46] Street, R. H., *The formal theory of monads*, J. Pure and Applied Algebra **2**, 1972, 149-168.

[47] Varacca, D., *The powerdomain of indexed valuations*, in *Proc. LICS 2002*, IEEE Computer Society, 2002, 299-310.

[48] Wadler, P., *Comprehending monads*, Mathematical Structures in Computer Science **2**, 1992, 461-493.