COMBINING COMPUTATIONAL EFFECTS: COMMUTATIVITY AND SUM

Martin Hyland
Department of Mathematics
University of Cambridge
Cambridge CB3 0WB, England
M.Hyland@dpmms.cam.ac.uk

Gordon Plotkin and John Power*
Laboratory for the Foundations of Computer Science
University of Edinburgh
King's Buildings, Edinburgh EH9 3JZ, Scotland
gdp@dcs.ed.ac.uk, ajp@dcs.ed.ac.uk

Abstract We seek a unified account of modularity for computational effects, using the notion of enriched Lawvere theory, together with its relationship with strong monads, to reformulate Moggi's paradigm for modelling computational effects. Effects qua theories are then combined by appropriate bifunctors (on the category of theories). We give a theory of the commutative combination of effects, which in particular yields Moggi's side-effects monad transformer. And we give a theory for the sum of computational effects, which in particular yields Moggi's exceptions monad transformer.

Keywords: Computational effects, enriched Lawvere theories, commutative combination, sum.

Introduction

We seek a unified account of modularity for computational effects. More precisely, we seek a mathematical theory that supports the combining of computational effects such as nondeterminism, probabilistic nondeterminism, side-effects, exceptions, interactive input/output, and continuations. Ideally, we should like to develop mathematical operations, together with associated rel-

---

*This work has been done with the support of EPSRC grant GR/M56333, a British Council grant and the COE budget of Japan.

R. Baeza-Yates et al. (eds.), *Foundations of Information Technology in the Era of Network and Mobile Computing* © Springer Science+Business Media New York 2002

evant theory. There is more than one such operation as, for example, the combination of side-effects and nondeterminism is of a different nature to the combination of side-effects and exceptions, and, further, as one is sometimes interested in different ways to combine even the same pair of effects. So we seek to find and develop what we expect will be a small number of computationally and mathematically natural such combining operations. We do not address continuations at all in this paper. But, with that caveat, the paper is devoted to two such ways of combining effects: that of combining them commutatively, as we shall see holds for combining side-effects with all effects we know other than exceptions; and that of taking their sum, as we shall see holds for combining exceptions with all other effects.

In order to give such operations, we first need a unified way to model the various computational effects individually. In this, we start by following Eugenio Moggi, who, in [15, 17], gave a unified category theoretic account of computational effects, which he called notions of computation. He modelled each effect by means of a strong monad T on a base category C with finite products. The monads corresponding to the effects listed above are given by a powerdomain [18], a probabilistic powerdomain [9, 10], and the monads $(S \times -)^S$, $-+E, TX = \mu Y.(O \times Y + Y^I + X)$, and R^{R^-} respectively [15, 16, 17], assuming C has appropriate additional structure; the set S of *states* is typically analysed as V^{Loc} where V is a set of *values* and Loc is a set of *locations.* Moggi's unified approach has proved useful, especially in functional programming [2].

Strong monads in hand, we first seek a binary operation that, to each pair of strong monads (T, T'), yields a new strong monad $T \otimes T'$, such that, in the case where $T' = (S \times -)^S$, the monad $T \otimes T'$ is $T(S \times -)^S$, which computational experience tells us is the natural combination of side-effects with all effects we know other than exceptions. So we ask: can we give a mathematical theory yielding such an operation on a pair of strong monads? Modulo a few side conditions, the answer is yes; we make fundamental use of the correspondence between strong monads and a generalised notion of Lawvere theory in order to provide it [23]. That correspondence is computationally natural and is already implicit in our previous work on computational effects [20, 21, 22]. We are unaware of any direct justification for the existence of $T \otimes T'$.

Here, we are following an algebraic programme that shifts focus away from monads to the study of natural operations that yield the required effects (see [20, 21] for other recent work along these lines), with the monads then corresponding to natural theories for these operations [22]. For instance, rather than emphasise the side-effects monad $(S \times -)^S$, we emphasise the operations *lookup* and *update* associated with side-effects, and the equations that relate them [22]. In the case where $S = V^{Loc}$, *lookup* can be considered as a *Loc*-indexed family of V-ary operations, and *update* as a $Loc \times V$-indexed family; the idea is that $lookup_l(x)$ proceeds with x_v if the contents of l is v and $update_{\langle l,v \rangle}(y)$ proceeds with y, having updated l with v. Again, rather than emphasise the powerdomain P, we emphasise the operation of nondeterministic choice $\vee$ with its equations for associativity, symmetry, and idempotence [7, 19].

This change in emphasis, supported by the correspondence between strong monads and enriched Lawvere theories in [23] (and see the expository [25]), is computationally natural for all the examples of computational effects listed above except for continuations [22]; in that case one can still make a formal change in emphasis, but it seems computationally unnatural, and we believe continuations should be treated separately.

Having reformulated our account of computational effects in terms of enriched Lawvere theories, we can reformulate our question to read: is there a mathematical theory yielding an operation that to each pair (L, L') of enriched Lawvere theories, gives a new enriched Lawvere theory $L \otimes L'$, such that, if L' is the enriched Lawvere theory associated with side-effects, the new enriched Lawvere theory corresponds to $T(S \times -)^S$, where L corresponds to T? The answer is yes, it is remarkably natural, and, in various guises, forms of it have existed since the 1960's [4, 13]. It is known as the *tensor product* of theories and simply amounts to taking the operations of both theories and demanding that they commute with each other, while retaining the equations of both. For instance, in the case where $S = V^{Loc}$, combining side-effects with nondeterminism, if there were three values, one would have the equation

$$\mathit{lookup}_l(x_1 \vee y_1, x_2 \vee y_2, x_3 \vee y_3) = \mathit{lookup}_l(x_1, x_2, x_3) \vee \mathit{lookup}_l(y_1, y_2, y_3)$$

In a functional language with references and nondeterminism this would induce the program equivalence:

$$\textbf{let } x \textbf{ be } !y \textbf{ in } (M \textbf{ or } N) \equiv (\textbf{let } x \textbf{ be } !y \textbf{ in } M) \textbf{ or } (\textbf{let } x \textbf{ be } !y \textbf{ in } N)$$

where $!M$ is the dereferencing operator and M **or** N is non-deterministic choice. There is a similar commutation equation for *update* and $\vee$, with a corresponding induced program equivalence. A recent reference for mathematical theory that supports this construction is [8], for which this is a leading example.

Having studied the commutative combination of effects, we turn to their sum. The natural combination of side-effects with exceptions is not their commutative combination. Exceptions combines with all other computational effects by taking the sum of the two theories: one has the operations for exceptions together with all operations for the other effect subject to all its equations, with no further equations. We shall show that the sum of theories yields Moggi's exceptions monad transformer, taking a monad T to the monad $T(- + E)$.

Of course, one typically combines more than two effects, so the operations we define may be used several times. For instance, to combine partiality, side-effects and nondeterminism, one can first combine partiality and semilattices by sum, then combine the result with side-effects by commutativity; similarly for partiality, side-effects and interactive input/output.

The published work most closely related to ours is that of Moggi and Cenciarelli on monad transformers. They defined a monad transformer to be a function from the set of strong monads on a category C with finite products to itself [2, 3]. The monad transformer for side-effects takes a monad T to the monad $T(S \times -)^S$, assuming C is cartesian closed. To model the combination of nondeterminism with side-effects, one would apply the side-effects monad transformer to a powerdomain P, yielding the monad $P(S \times -)^S$. So the resulting monad agrees with ours, as it must, but we have an associated mathematical theory: the question we pose could equally be posed to ask how one might derive the side-effects monad transformer from the side-effects monad *qua* monad, but the work on monad transformers to date has not answered that. Moreover, our work involves no asymmetry: there seems no a priori reason why the combination of side-effects with nondeterminism should be achieved by applying the side-effects monad transformer to the nondeterminism monad rather than vice-versa. And in the case of exceptions, the side-effects monad transformer does not even give the required result for the usual interpretation of the combination.

There is also relevant unpublished work by Paul Levy. He has observed that the sum of any monad T with that for exceptions $- + E$ is given by $T(- + E)$. He has also defined a notion of commutative combination of monads and shown that $T(S \times -)^S$ is the commutative combination of T and $(S \times -)^S$ with that definition. His construction does not exist for all pairs of monads, and he has not developed accompanying theory; but when the monads have rank, his definition agrees with ours.

Other than the work on monad transformers, the other main attempt we know to account for the combination of side-effects with other computational effects has been the development of dyads [24] which amount to a decomposition of the side-effects monad into strictly more primitive structure. Dyads come equipped with a notion of Kleisli category, in which one may model the computational λ-calculus, and have been integrated with *Freyd*-structure, which models a delicate feature of contexts arising with side-effects or exceptions, where the order of evaluation is crucial. The relationship between the two notions remains to be investigated.

The paper is organised as follows. We do not have space here to explain the general enriched setting. So in each section we investigate the unenriched case, which largely amounts to the situation where computational effects are modelled in *Set*, then we briefly remark on its enrichment, especially to ωCpo (whose objects are ω-cpos, partial orders with least upper bounds of increasing ω-chains, and whose morphisms are continuous functions, i.e., maps of partial orders that preserve the least upper bounds). In Section 1, we describe the relationship between monads and Lawvere theories, and explain how the latter appear in our leading examples. In Section 2, we explore the commutative combination of Lawvere theories. In Section 3, we show that the commutative combination of side-effects with any other Lawvere theory gives the outcome we seek. And in Section 4, we develop a theory for the sum of Lawvere theories and explain how this gives rise to the exceptions monad transformer.

A clear omission from this paper is the study of distributivity: this seems to be the main other way in which computational effects combine. One has distributivity of one set of operations over the other in combining nondeterminism with probabilistic nondeterminism [14], and one has distributivity of each set of operations over the other in combining internal and external nondeterminism [6]. Another important question concerns the combination of effects with local state [22] (this paper only concerns global state). In [22] local state is specified using an additional operation *block* together with additional equations. But it is unclear yet how best to integrate it with enriched Lawvere theories, let alone consider combinations with other effects. We have also not considered the combination of interactive input/output with other effects in this paper, but we believe that sum is the operation of primary interest there. Finally, we have not considered the relationship of all these effects with that of continuations; this will be substantially different as the continuations monad does not have a rank.

1. Monads and Lawvere theories

For simplicity of exposition, we start by restricting our attention to the base category *Set*. The side-effects monad is then the monad $(S \times -)^S$ for a set of states S. We impose the natural restriction that S is a countable set, and in case $S = V^{Loc}$ restrict V to be countable and Loc to be finite (when dealing with local state with its need for unboundedly many locations, one can use a presheaf semantics [22]).

The side-effects monad is then of *countable rank*, which means that, in a precise sense, it is of bounded size [11]. The category of monads with countable rank is equivalent to the category of countable Lawvere theories as we shall outline. So, in principle, the side-effects monad can equally be seen as a countable Lawvere theory, and that turns out to be a computationally natural way in which to see it. All our mathematics generalises to arbitrary rank, but, as all our examples are of countable rank, we naturally restrict our exposition to that case.

Let $\aleph_1$ denote a skeleton of the category of countable sets and all functions between them. So $\aleph_1$ has an object for each natural number n and an object for $\aleph_0$. Up to equivalence, $\aleph_1$ is the free category with countable coproducts on 1. So, in referring to $\aleph_1$, we implicitly make a choice of the structure of its countable coproducts.

Definition 1 *A* countable Lawvere theory *consists of a small category L with countable products and a strict countable-product preserving identity-on-objects functor $I : \aleph_1^{op} \longrightarrow L$. A* model *of a countable Lawvere theory L in any category C with countable products is a countable-product preserving functor $M : L \longrightarrow C$.*

For any countable Lawvere theory L and any category with countable products C, we thus have the category $Mod(L,C)$ of models of L in C. There is a canonical forgetful functor $U : Mod(L,C) \longrightarrow C$, and, when $C = Set$, this forgetful functor has a left adjoint, exhibiting $Mod(L, Set)$ as equivalent to the category T_L-Alg for the induced monad T_L on Set.

Conversely, given a monad T with countable rank on Set, the category $Kl(T)^{op}_{\aleph_1}$ determined by restricting $Kl(T)$ to the objects of $\aleph_1$ is a countable Lawvere theory L_T, and the functor from T-Alg to $Mod(L_T, Set)$ induced by the restriction is an equivalence of categories. The following theorem appears in enriched form, as we ultimately require it, in [23].

Theorem 1 *The construction sending a countable Lawvere theory* L *to* T_L *together with that sending a monad* T *with countable rank to* L_T *induce an equivalence of categories between the category of countable Lawvere theories and the category of monads with countable rank on Set. Moreover, the comparison functor exhibits an equivalence of the categories* $Mod(L, Set)$ *and* T_L*-Alg.*

So, in principle, the side-effects monad can be described as a countable Lawvere theory. The usual way in which to define countable Lawvere theories is by means of sketches, with the Lawvere theory given freely on the sketch. To give a sketch amounts to giving operations and equations, here the operations being allowed to be of countable arity: Barr and Wells' book [1] treats sketches in loving detail. A sketch, and hence the countable Lawvere theory, corresponding to the side-effects monad is essentially given in [22] and is easy to describe:

Example 1 *The countable Lawvere theory* L_S *for side-effects (when* $S = V^{Loc}$*) is the free countable Lawvere theory generated by operations* $lookup : V \longrightarrow Loc$ *and* $update : 1 \longrightarrow Loc \times V$ *subject to the seven natural equations listed in [22], four of them specifying interaction equations for lookup and update and three of them specifying commutation equations. Note the use of the targets Loc and* $Loc \times V$ *to handle indexing at the Lawvere theory level.*

Proposition 1 *[22] For any category* C *with countable products and countable coproducts, the canonical comparison functor from* $Mod(L_S, C)$ *to* T*-Alg is an equivalence of categories, where* T *is the monad on* C *defined by* $TX = (\Sigma_S X)^S$.

Example 2 *Ignoring partiality, 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.*

Example 3 *The countable Lawvere theory* L_P *for (binary) probabilistic nondeterminism is that freely generated by* $[0, 1]$*-many binary operations* $+_r : 2 \longrightarrow 1$ *subject to the equations for associativity, commutativity and idempotence in [5].*

Example 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*. So, interactive input/output is more directly modelled by the countable Lawvere theory than by the corresponding monad* $TX = \mu Y.(O \times Y + Y^I + X)$.

Example 5 *The countable Lawvere theory* L_E *for exceptions is the free countable Lawvere theory generated by the operation* $raise : 0 \longrightarrow E$*, where* E *is a countable set of* exceptions.

Of course, *Set* is not the category of primary interest in denotational semantics. One is more interested in ωCpo, and variants, in order to model recursion. The relationship between countable Lawvere theories and countable monads extends without fuss to one between countable enriched Lawvere theories and countable strong monads on such categories [23].

Example 6 *The countable Lawvere ωCpo-theory $L_\perp$ for partiality is the theory freely generated by a nullary operation $\perp : 0 \longrightarrow 1$ subject to the condition that there is an inequality*

$$\begin{tikzcd}
1 \arrow[r] \arrow[dr, "\geq", "id"'] & 0 \arrow[d, "\perp"] \\
 & 1
\end{tikzcd}$$

where the unlabelled map is the unique map determined because 0 *is the terminal object of $V_{\aleph_1}^{op}$. A model of $L_\perp$ in ωCpo is exactly an ω-cpo with least element.*

We have already introduced a countable Lawvere theory L_P corresponding to a powerdomain: it is the countable Lawvere theory for a semilattice. We overload notation a little here by also using the notation L_P to denote the countable Lawvere ωCpo-theory for a semilattice: the generators and equations are the same, but the ωCpo-theory has more objects as there are countably presentable ω-cpos other than flat ones, and these additional objects generate additional maps. But the countable Lawvere ωCpo-theory for a semilattice is still just the free countable Lawvere ωCpo-theory on the countable Lawvere theory for a semilattice.

This definition allows us to make immediate reference to the sum of effects that we shall define later. Using the terminology we shall define, we can therefore describe the countable Lawvere ωCpo-theory for nondeterminism.

Example 7 *The countable Lawvere ωCpo-theory for nondeterminism is given by the sum of the countable Lawvere ωCpo-theories L_N for a semilattice and $L_\perp$ for partiality.*

The combination of partiality with other effects is typically given by sum. But that is not always the case: the combination with side-effects is given by taking the commutative combination, which we define in the next section.

Another non-trivial example of a computationally natural countable Lawvere ωCpo-theory is given by probabilistic nondeterminism [5, 9, 10, 22]. This includes L_P and $L_\perp$ as well as an infinitary axiom.

2. The commutative combination of effects

In this section, we define the commutative combination $L \otimes L'$ of countable Lawvere theories L and L' and develop mathematical theory in support of the definition of this tensor product.

The category $\aleph_1$ not only has countable coproducts, but also has finite products, which we denote by $A \times A'$. The object $A \times A'$ may also be seen as the coproduct of A copies of A', so, given an arbitrary map $f' : A' \longrightarrow B'$ in a countable Lawvere theory, it is immediately clear what we mean by the morphism $A \times f' : A \times A' \longrightarrow A \times B'$. We define $f \times A'$ by conjugation, and, in the following, we suppress the canonical isomorphisms.

Definition 2 *Given countable Lawvere theories L and L', the countable Lawvere theory $L \otimes L'$ is defined by the universal property of having maps of countable 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*

$$\begin{tikzcd}
A \times A' \arrow[r, "A \times f'"] \arrow[d, "f \times A'"'] & A \times B' \arrow[d, "f \times B'"] \\
B \times A' \arrow[r, "B \times f'"'] & B \times B'
\end{tikzcd}$$

Theorem 2 *The construction $\otimes$ extends canonically to a symmetric monoidal structure on the category of countable Lawvere theories. Moreover, for any small category C with countable products, there is a coherent equivalence of categories between $Mod(L \otimes L', C)$ and $Mod(L, Mod(L', C))$.*

Example 8 *Letting L_S be the countable Lawvere theory for side-effects, if C has countable products and countable coproducts, we have seen that $Mod(L_S, C)$ is equivalent to the category T-Alg for the monad $TX = (\Sigma_S X)^S$ on C. For any countable Lawvere theory L, the category $Mod(L, Set)$ is always complete and cocomplete, so has countable products and countable coproducts. So, by the theorem, $Mod(L_S \otimes L, Set)$ is equivalent to T-Alg for $TX = (\Sigma_S X)^S$ taken as a monad on $Mod(L, Set)$.*

3. The commutative combination of side-effects with other effects

Here, we study the commutative combination of side-effects with other computational effects in more detail. Our central result is as follows.

Theorem 3 *Let L_S denote the countable Lawvere theory for side-effects (where $S = V^{Loc}$) and let L denote any countable Lawvere theory. Then the monad $T_{L_S \otimes L}$ is isomorphic to $T_L(S \times -)^S$.*

Proof 1 *We have seen in preceding sections that $Mod(L_S, Mod(L, Set))$ is equivalent to T-Alg, where T is the monad on $Mod(L, Set)$ given by $TX = (\Sigma_S X)^S$. The category $Mod(L, Set)$ is equivalent to T_L-Alg. So we denote the canonical adjunction by $F_L \dashv U_L : Mod(L, Set) \longrightarrow Set$. Right adjoints preserve products, left adjoints preserve coproducts, and a coproduct*

$\Sigma_Y X$ in Set is given by $Y \times X$. So the monad $T_{L_S \otimes L}$, which, by our main theorem, is the monad determined by the composite forgetful functor from T-Alg to Set, must be given by $T_{L_S \otimes L}X = U_L(\Sigma_S F_L X)^S = T_L(S \times X)^S$ as required.

We do not require rank, in particular countability, for this result. We could define a notion of theory that does not involve a rank, retain a correspondence with strong monads, and make the commutative combination of the theorem, but the general theory becomes more complicated because commutative combinitions of such theories do not always exist.

This result shows that, under the hypotheses of the theorem, our theory of the commutative combination of computational effects agrees with Moggi's definition of the side-effects monad transformer. In particular, this accounts for the interaction between side-effects and nondeterminism, and in doing so, the theory yields not just an object of values for the combination but a description of natural operations and the way in which they interact with each other, and it follows immediately from the definition of $\otimes$ that one does not lose any of the equations for either nondeterminism or side-effects with which one began. It is also interesting to note that the side-effects theory for $S = V^{Loc}$ is the *Loc*-fold tensor product of the side-effects theory for $S = V$.

4. The sum of effects

Finally, we turn to the sum of effects, our leading example being given by the combination of exceptions with all other computational effects we have considered, such as side-effects, nondeterminism, and interactive input/output.

Theorem 4 *[12, 23] The category of countable Lawvere theories is cocomplete.*

Theorem 5 *Given a set E, if L_E is the countable Lawvere theory for E nullary operations, and if L is any countable Lawvere theory, then T_{L_E+L} is given by the theory $T_L(- + E)$.*

Proof 2 *The category $T_L(- + E)$-Alg is isomorphic to T'_L-Alg, where T'_L is the monad on $(- + E)$-Alg determined by lifting T_L, using the canonical distributive law of $- + E$ over T_L. By direct calculation, one can see that the latter category is in turn isomorphic to $(T_L + (- + E))$-Alg: a T'_L-algebra consists of a set X together with E elements of X and a T_L-structure on X, i.e., a $(T_L + (- + E))$-algebra.*

This result explains how the exceptions monad transformer, sending a monad T to the composite $T(- + E)$, arises: take the disjoint union of the two sets of operations and retain the equations for T. And it provides our usual theory of coproducts, such as its associativity and commutativity, and its interaction with other operations.

Definition 3 *Given a countable Lawvere theory L and a category C with countable products, denote by $Mod^*(L, C)$ the identity-on-objects/fully faithful factorisation of the forgetful functor $U : Mod(L, C) \longrightarrow C$.*

Theorem 6 *There is a natural equivalence between* $Mod^*(L + L', C)$ *and* $Mod^*(L, Mod^*(L', C))$.

As in previous sections, the analysis of this section all enriches without fuss, with the sum again being the correct operation in the enriched setting.

References

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

[2] N. Benton, J. Hughes, and E. Moggi, *Monads and Effects*, APPSEM '00 Summer School, 2000.

[3] P. Cenciarelli and E. Moggi, *A Syntactic Approach to Modularity in Denotational Semantics*, CWI Technical Report, 1983.

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

[5] R. Heckmann, Probabilistic Domains, in *Proc. CAAP '94*, LNCS, Vol. 136, pp. 21-56, Berlin: Springer-Verlag, 1994.

[6] M. C. B. Hennessy, *Algebraic Theory of Processes*, Cambridge, Massachusetts: MIT Press, 1988.

[7] M. C. B. Hennessy and G. D. Plotkin, Full Abstraction for a Simple Parallel Programming Language, in *Proc. MFCS '79* (ed. J. Beĉvâř), LNCS, Vol. 74, pp. 108-120, Berlin: Springer-Verlag, 1979.

[8] J. M. E. Hyland and A. J. Power, Pseudo-Closed 2-Categories and Pseudo-Commutativities, *J. Pure Appl. Algebra*, to appear.

[9] C. Jones, *Probabilistic Non-Determinism*, Ph.D. Thesis, University of Edinburgh, Report ECS-LFCS-90-105, 1990.

[10] C. Jones and G. D. Plotkin, A Probabilistic Powerdomain of Evaluations, in *Proc. LICS '89*, pp. 186–195, Washington: IEEE Press, 1989.

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

[12] G. M. Kelly and A. J. Power, Adjunctions whose Counits are Coequalizers, and Presentations of Finitary Enriched Monads, *J. Pure Appl. Algebra*, Vol. 89, pp. 163–179, 1993.

[13] E. G. Manes, *Algebraic Theories*, Graduate Texts in Mathematics, Vol. 26, New York: Springer-Verlag, 1976.

[14] M. W. Mislove, Nondeterminism and Probabilistic Choice: Obeying the Laws, in *International Conference on Concurrency Theory*, pp. 350–364, URL: http://www.math.tulane.edu/ mwm, 2000.

[15] E. Moggi, Computational Lambda-Calculus and Monads, in *Proc. LICS '89*, pp. 14–23, Washington: IEEE Press, 1989.

[16] E. Moggi, *An Abstract View of Programming Languages*, University of Edinburgh, Report ECS-LFCS-90-113, 1989.

[17] E. Moggi, Notions of Computation and Monads, *Inf. and Comp.*, Vol. 93, No. 1, pp. 55–92, 1991.

[18] G. D. Plotkin, A Powerdomain Construction, *SIAM J. Comput.* Vol. 5, No. 3, pp. 452–487, 1976.

[19] G. D. Plotkin, *Domains*, URL: http://www.dcs.ed.ac.uk/home/gdp, 1983.

[20] G. D. Plotkin and A. J. Power, Adequacy for Algebraic Effects, in *Proc. FOSSACS 2001* (eds. F. Honsell and M. Miculan), LNCS, Vol. 2030, pp. 1–24, Berlin: Springer-Verlag, 2001.

[21] G. D. Plotkin and A. J. Power, Semantics for Algebraic Operations (extended abstract), in *Proc. MFPS XVII* (eds. S. Brookes and M. Mislove), ENTCS, Vol. 45, Amsterdam: Elsevier, 2001.

[22] G. D. Plotkin and A. J. Power, Notions of Computation Determine Monads, in *Proc. FOSSACS 2002* (eds. M. Neilsen and U. Engberg), LNCS, Vol. 2303, pp. 342–356, Berlin: Springer-Verlag, 2002.

[23] A. J. Power, Enriched Lawvere Theories, in *Theory and Applications of Categories*, pp. 83–93, 2000.

[24] A. J. Power and E. P. Robinson, Modularity and Dyads, in *Proc. MFPS XV* (eds. S. Brookes, A. Jung, M. Mislove and A. Scedrov), ENTCS Vol. 20, Amsterdam: Elsevier, 1999.

[25] E. Robinson, *Variations on Algebra: Monadicity and Generalisations of Equational Theories*, to appear, URL: http://www.dcs.qmul.ac.uk/ edmundr/publications.html, 2001.