Combining computational effects: commutativity and sum

Martin Hyland,[1] Gordon Plotkin and John Power[2,⋆]

[1] Department of Mathematics, University of Cambridge, Cambridge, England. Email: M.Hyland@dpmms.cam.ac.uk

[2] Laboratory for the Foundations of Computer Science, University of Edinburgh, King's Buildings, Edinburgh, EH9 3JZ, Scotland. Email:gdp@dcs.ed.ac.uk, ajp@dcs.ed.ac.uk. Tel: +44 131 650 5159. Fax: +44 131 667 7209.

Abstract. We begin to develop a unified account of modularity for computational effects. We use the notion of enriched Lawvere theory, together with its relationship with strong monads, to reformulate Moggi's paradigm for modelling computational effects; we emphasise the importance here of the operations that induce 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 (an application is the combination of side-effects with nondeterminism). And we give a theory for the sum of computational effects, which in particular yields Moggi's exceptions monad transformer (an application is the combination of exceptions with other effects).

1 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 relevant theory, that, given any pair of the above effects, yields their combination. There is more than one such operation as, for example, the combination of state and nondeterminism is of a different nature to the combination of state 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. This paper is devoted to two such ways of combining effects: that of combining them commutatively, as we shall see holds for combining state with all effects we know other than continuations and exceptions; and that of taking their sum, as we shall see holds for combining exceptions with all effects other than continuations.

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

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 [17, 19], 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 [20], a probabilistic powerdomain [11, 12], and the monads $(S\times-)^S$, $-+E$, $TX = \mu Y.(O \times Y + Y^I + X)$, and R^{R^-} respectively [17–19], 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 state with all effects we know other than continuations and 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, but we must make fundamental use of the correspondence between strong monads and a generalised notion of Lawvere theory in order to provide it [25]. That correspondence is computationally natural and is already implicit in our previous work on computational effects [22–24]. We are unaware of any direct justification for the existence of $T \otimes T'$.

Recently, we have followed an algebraic programme that shifts focus away from monads to the study of natural operations that yield the required effects [22, 23], with the monads then corresponding to natural theories for these operations [24]. For instance, rather than emphasise the side-effects monad $(S \times -)^S$, one emphasises the operations *lookup* and *update* associated with state, and the equations that relate them [24]. 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, one emphasises the nondeterministic operation $\vee$ with its equations for associativity, symmetry, and idempotence [7, 21]. This change in emphasis, supported by the correspondence between strong monads and enriched Lawvere theories in [25] (and see the expository [29]), is computationally natural for all the examples of computational effects listed above except for continuations [24]; 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, 15], and is known as the *tensor product* of theories. It 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 state with nondeterminism, if there were three values, one would have the equation

$$lookup_l(x_1 \vee y_1, x_2 \vee y_2, x_3 \vee y_3) = lookup_l(x_1, x_2, x_3) \vee lookup_l(y_1, y_2, y_3)$$

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. We prove, using the first main result of [24], that the monad corresponding to this enriched Lawvere theory is indeed given by $T(S \times -)^S$. Recent references for mathematical theory that supports this construction are [8–10], 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. Equivalently, it is not given by applying the side-effects monad transformer to the exceptions monad. Exceptions combines with all other computational effects we know (other than continuations) 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, taking one theory to be that for exceptions, the sum of theories agrees with Moggi's exceptions monad transformer, taking a monad T to $T(- + E)$.

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 state by commutativity; similarly for partiality, side-effects and interactive input/output.

The work most closely related to ours is that of Moggi and his collaborators on monad transformers. Moggi and Cenciarelli 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 state 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.

Other than the work on monad transformers, the other main attempt we know to account for the combination of state with other computational effects has been the development of dyads [26–28] 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 state 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. Throughout the paper, in each section, we first investigate the unenriched case, which largely amounts to the situation where computational effects are modelled in *Set*, then we explain the more general situation that includes base categories such as ωCpo, the category of primary interest in denotational semantics (its objects are ω-cpos, partial orders with least upper bounds of increasing ω-chains, and its morphisms are continuous functions, i.e., maps of partial orders that preserve the least upper bounds). In Section 2, we describe the relationship between monads and Lawvere theories, and explain how the latter appear in our leading examples. In Section 3, we explore the commutative combination of Lawvere theories. In Section 4, we show that the commutative combination of state with any other Lawvere theory gives the outcome we seek. And in Section 5, we develop a theory for the sum of Lawvere theories and explain how this gives rise to the exceptions monad transformer. We end with an appendix the fundamental 2-categorical theory that underlies our main results.

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 [16], and one has distributivity of each set of operations over the other combining internal and external nondeterminism [6]. Another important question concerns the combination of effects with local state [24] (this paper only concerns global state). In [24] 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.

2 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 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, we would use a presheaf semantics rather than allowing Loc to be infinite).

The side-effects monad is then of *countable rank*, which means that, in a precise sense, it is of bounded size [13]: a precise definition of the notion of countable rank is complicated, so we simply remark that all monads we mention in this paper except that for continuations are of countable rank. 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 is a computationally natural way in which to see it, as we shall explain.

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 $I : \aleph_1^{op} \longrightarrow L$.*

Definition 2. *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, with maps given by all natural transformations: the naturality condition implies that the maps respect countable product structure. 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 $Kl(T)_{\aleph_1}^{op}$ 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.

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 the category $Mod(L, Set)$ as equivalent to the category 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 [24] and is easy to describe as follows:

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 [24], 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.

The following result is a restatement of the first main result of [24]. It is fundamental for the proof of the main result of this paper. Observe that it refers to target categories other than Set.

Proposition 1. *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 $TX = (\Sigma_S X)^S$.*

We shall now consider how examples other than side-effects appear as countable Lawvere theories.

Example 2. Ignoring partiality, the countable Lawvere theory L_P corresponding to a powerdomain 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 $L_{I/O}$ 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 4. 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.*.

Details of the examples of exceptions and interactive input/output, and in less detail probabilistic nondeterminism, can be readily understood from [24].

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. The least obvious point to note here is that the notion of countable product of a single generator does not generalise most naturally to a notion of countable product but rather to a notion of countable *cotensor*.

The notion of cotensor is the most natural enrichment of the notion of a power-object. Given an object A of a category C and given a set X, the power A^X satisfies the defining condition that there is a bijection of sets

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

natural in B. Enriching this, given an object A of a V-category C and given an object X of V, the cotensor A^X satisfies the defining condition that there is an isomorphism in V

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

V-natural in B. For instance, taking V to be *Poset*, this allows us not only to consider objects such as $A \times A$ in a locally ordered category, but also to consider objects such as $A^{\leq}$. This possibility allows us, in describing theories, to consider a greater range of arities and to incorporate inequations in the context of an elegant, coherent body of mathematics. In general, we still have all countable products of objects in an countable Lawvere V-theory, but we also have a little more.

Given a category V that is locally countably presentable as a cartesian closed category, for instance ωCpo, one defines $V_{\aleph_1}$ to be a skeleton of the full sub-V-category of V determined by the countably presentable objects of V. It is equivalent to the free V-category with countable tensors on 1 [13, 25].

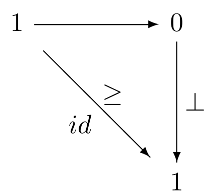
Definition 3. *A* countable Lawvere V-theory *is a small V-category L with countable cotensors together with a strict countable-cotensor preserving identity-on-objects V-functor $I : V_{\aleph_1}^{op} \longrightarrow L$. A* model *of L in a V-category C with countable cotensors is a countable-cotensor preserving V-functor $M : L \longrightarrow C$.*

The constructions of T_L from L and of L_T from T are routine generalisations of those in the unenriched setting, and the central results for the unenriched setting generalise routinely too. To give a V-enriched V-monad is equivalent to giving a strong monad on V, so in order to make the comparison with Moggi's definition a little more direct, we express the main abstract result in terms of strong monads.

Theorem 2. *If V is locally countably presentable as a cartesian closed category, the constructions of T_L from L and of L_T from T induce an equivalence of categories between the category of countable Lawvere V-theories on V and the category of strong monads on V with countable rank. Moreover, the comparison V-functor exhibits the V-category $Mod(L, V)$ as equivalent to the V-category T_L-Alg.*

For an example of a countable Lawvere V-theory that does not arise freely from an unenriched countable Lawvere theory, let V be the ωCpo, and consider a countable Lawvere theory for partiality.

Example 5. 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



where the unlabelled map is the unique map determined because 0 is the initial object of $V_{\aleph_1}$ and therefore 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 as follows:

Example 6. The countable Lawvere ωCpo-theory L_N for nondeterminism is given by the sum of the countable Lawvere ωCpo-theories L_P 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, 11, 12]. More detail appears in [24], albeit in the mathematical terms of [14].

3 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. We move immediately to the central definition of the section, for base category *Set*.

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 4. *Given countable Lawvere theories L and L', the countable Lawvere theory $L \otimes L'$ is defined to be the countable Lawvere theory generated by the disjoint union of all operations of L and all operations of L' subject to all equations of L, all equations of L', and commutativity of all operations of L with respect to all operations of L', i.e., given $f : A \longrightarrow B$ in L and $f' : A' \longrightarrow B'$*

in L', we demand commutativity of the diagram

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

We shall systematically investigate the commutative combination of state with other effects in the next section. In the previous section, we made use of an enriched commutative operation generalising $\otimes$ when we combined side-effects with partiality. We shall spell out the enriched version of $\otimes$ below. But first we develop the abstract theory in the unenriched case.

Proposition 2. *The construction $\otimes$ extends canonically to a symmetric monoidal structure on the category of countable Lawvere theories.*

A proof for this proposition is elementary. The result gives some indication of the definiteness of the operation sending (L, L') to $L \otimes L'$, but not much: there are typically many symmetric monoidal structures on categories, such as finite product or finite coproduct, typically with many others too. But what is much less common is a closedness condition. Categories typically have few if any closed structures on them. Moreover, the analysis of a closed structure corresponding to $\otimes$ is essential to the proof of our main theorem about the combination of state with other effects. But to have a closed structure, we would require that, for any category C with countable products, the category $Mod(\aleph_1^{op}, C)$ is isomorphic to C, as that is required to give the isomorphism $C^I \cong C$ that must hold in any closed structure; but we do not have that condition here because, in fact, the category $Mod(\aleph_1^{op}, C)$ is equivalent but not isomorphic to C. But we do have a *pseudo-closed* structure, a definition of which is in Appendix A, together with the relevant, delicate 2-categorical analysis.

Theorem 3. *The construction $L \otimes L'$ on countable Lawvere theories extends to a coherent pseudo-monoidal pseudo-closed structure on the 2-category of small categories with countable products, and, in particular, 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 7. 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 $Mod(L, Set)$.

The analysis of this section extends readily to the enriched setting as follows. The V-category $V_{\aleph_1}$ not only has countable tensors but also has finite products, just as the ordinary category $\aleph_1$ not only has countable coproducts but also has finite products. Our analysis of $A \times f'$ in the unenriched setting extends routinely to the enriched setting, except here, of course, we must express the analysis in terms of the object $L(A', B')$ of V rather than in terms of an arrow $f' : A' \longrightarrow B'$. The key fact is that the cotensor $(A^X)^Y$ is canonically isomorphic to the cotensors $A^{(}X \times Y)$ and $(A^Y)^X$. Consistently with this, we must express the commutativity condition of the theorem in terms of homobjects of V rather than in terms of arrows like f'.

Definition 5. *Let V be locally countably presentable as a cartesian closed category. Given countable Lawvere V-theories L and L', the countable Lawvere V-theory $L \otimes L'$ is defined to be the countable Lawvere V-theory generated by the disjoint union of $L(A, B)$ and $L'(A, B)$ for each (A, B), subject to all equations of L, all equations of L', and, suppressing canonical isomorphisms, commutativity of*

$$\begin{array}{ccc}
L(A,B) \times L'(A',B') & \longrightarrow & L(A \times B', B \times B') \times L'(A \times A', A \times B') \\
\downarrow & & \downarrow{\scriptstyle comp} \\
L(A \times A', B \times A') \times L'(B \times A', B \times B') & \xrightarrow[comp]{} & L(A \times A', B \times B')
\end{array}$$

Theorem 4. *Let V be locally countably presentable as a cartesian closed category. Then the construction $L \otimes L'$ on countable Lawvere V-theories is symmetric monoidal and extends to a coherent pseudo-monoidal pseudo-closed structure on the 2-category of small V-categories with countable cotensors, and, for any small V-category C with countable cotensors, there is a coherent equivalence of V-categories between $Mod(L \otimes L', C)$ and $Mod(L, Mod(L', C))$.*

4 The commutative combination of state 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 5. *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 $(S \times T_L -)^S$.*

Proof. 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.

The theorem generalises readily to the enriched setting, taking S to be any countably presentable object of V.

Theorem 6. *Let V be locally countably presentable as a cartesian closed category, and let S be a countably presentable object of V. Let L_S denote the countable Lawvere V-theory for side-effects, and let L denote any countable Lawvere V-theory. Then the monad $T_{L_S \otimes L}$ is isomorphic to $(S \times T_L -)^S$.*

Proof. The proof is essentially the same as for the unenriched case. One must observe that the proof of the first main theorem in [24] extends routinely to enrichment in a category that is locally countably presentable as a cartesian closed category, taking $Loc = 1$ and $V = S$.

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 state 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 $S = V$.

5 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. A succession of results support the construction of the sum of theories.

Theorem 7. *The category of countable Lawvere theories is cocomplete.*

This may be proved using the equivalence between countable Lawvere theories and monads on *Set* of countable rank, together with the analysis of [14]. We therefore know that the sum of countable Lawvere theories exists.

Theorem 8. *Given a set E, if L_E denotes the countable Lawvere theory for E nullary operations, and if L is any countable Lawvere theory, T_{L_E+L} is given by $T_L(- + E)$.*

Proof. 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: one simply takes the disjoint union of the two sets of operations and retains the equations for T. And this explanation brings with it our usual theory of coproducts, such as its associativity and commutativity, and its interaction with other operations.

The elegant 2-category theory underlying the commutative combination of effects does not seem relevant here, but we still do have a closedness result as follows:

Definition 6. *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 $U : Mod(L, C) \longrightarrow C$.*

So the objects of $Mod^*(L, C)$ are the models of L in C and the maps are just maps in C. We have the following theorem:

Theorem 9. *There is a natural equivalence between $Mod^*(L + L', C)$ and $Mod^*(L, Mod^*(L', C))$.*

A simple proof is given by heavy use of the correspondence between countable Lawvere theories and monads on *Set* with countable rank, with the knowledge that a $T + T'$-algebra consists of a set X together with both a T-structure and a T'-structure on it.

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. Beĉvář), LNCS, Vol. 74, pp. 108-120, Berlin: Springer-Verlag, 1979.
8. J. M. E. Hyland and A. J. Power, Pseudo-commutative monads, *Proc. MFPS XVII*, ENTCS Vol. 45, Amsterdam: Elsevier, 2001.
9. J. M. E. Hyland and A. J. Power, Two-dimensional linear algebra, *Proc. CMCS 2001* ENTCS Vol. 47, Amsterdam: Elsevier, 2001.
10. J. M. E. Hyland and A. J. Power, Pseudo-closed 2-categories and pseudo-commutativities, *J. Pure Appl. Algebra*, to appear.

11. C. Jones, *Probabilistic Non-Determinism*, Ph.D. Thesis, University of Edinburgh, Report ECS-LFCS-90-105, 1990.
12. C. Jones and G. D. Plotkin, A Probabilistic Powerdomain of Evaluations, in *Proc. LICS '89*, pp. 186–195, Washington: IEEE Press, 1989.
13. G. M. Kelly, *Basic Concepts of Enriched Category Theory*, Cambridge: Cambridge University Press, 1982.
14. 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.
15. E. G. Manes, *Algebraic Theories*, Graduate Texts in Mathematics, Vol. 26, New York: Springer-Verlag, 1976.
16. 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.
17. E. Moggi, Computational lambda-calculus and monads, in *Proc. LICS '89*, pp. 14–23, Washington: IEEE Press, 1989.
18. E. Moggi, *An abstract view of programming languages*, University of Edinburgh, Report ECS-LFCS-90-113, 1989.
19. E. Moggi, Notions of computation and monads, *Inf. and Comp.*, Vol. 93, No. 1, pp. 55–92, 1991.
20. G. D. Plotkin, A Powerdomain Construction, *SIAM J. Comput.* Vol. 5, No. 3, pp. 452–487, 1976.
21. G. D. Plotkin, *Domains*, (http://www.dcs.ed.ac.uk/home/gdp/), 1983.
22. 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.
23. 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.
24. G. D. Plotkin and A. J. Power, Notions of computation determine monads, to appear in FOSSACS '02, URL:http://www.dcs.ed.ac.uk/home/gdp/public ations/.
25. A. J. Power, Enriched Lawvere Theories, in *Theory and Applications of Categories*, pp. 83–93, 2000.
26. A. J. Power, Modularity in denotational semantics, in *Electronic Notes in Theoretical Computer Science*, Vol. 6 (eds. S. Brookes and M. Mislove), Amsterdam: Elsevier, 1997.
27. 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.
28. A. J. Power and G. Rosolini, A Modular Approach to Denotational Semantics, in *Proc. ICALP 98* LNCS 1443, pp. 351–362, 1998.
29. E. Robinson, *Variations on Algebra: monadicity and generalisations of equational theories*, to appear, URL: http://www.dcs.qmul.ac.uk/ edmundr/p ublications.html, 2001.

A Pseudo-commutativity and pseudo-closedness of CP

The simplest way we know to explain the extent to which we have a natural closed structure on the category of small categories with countable products is in terms of

2-monads on Cat as developed in [8, 10], cf also [9]. The 2-monad of interest to us is the 2-monad T_{cp} for which the 2-category of algebras, pseudo-maps, and 2-cells is the 2-category of small categories with countable products, functors that preserve countable products in the usual sense, and natural transformations.

Definition 7. *A symmetric pseudo-commutativity for a 2-monad T on Cat consists of a family of invertible natural transformations*

$$\begin{array}{ccccc} TA \times TB & \xrightarrow{t^*} & T(A \times TB) & \xrightarrow{Tt} & T^2(A \times B) \\ {\scriptstyle t}\downarrow & & \Downarrow \gamma_{A,B} & & \downarrow{\scriptstyle \mu_{A\times B}} \\ T(TA \times B) & \xrightarrow[Tt^*]{} & T^2(A \times B) & \xrightarrow[\mu_{A\times B}]{} & T(A \times B) \end{array}$$

natural in A and B and subject to coherence with respect to the symmetry of Cat and one coherence axiom with respect to each of the strength, unit, and multiplication of T.

The monad T_{cp} has a unique symmetric pseudo-commutativity. The first main definition of [10] gives a notion of *pseudo-closed* structure for a 2-category: it is almost as strong as closed structure, but one needs to relax the definition of closed structure just a little in order to account for the distinctions between preservation and strict preservation of structure such as countable product structure: the reason, in our setting, that we do not quite have a closed structure is that, given a category C with countable products, the category $Mod(\aleph_1^{op}, C)$ is equivalent but not isomorphic to C. We do not spell out the detailed definition of pseudo-closed 2-category here. The main result of [10] (see [8] for a formulation directed more towards a computer science audience) is as follows:

Theorem 10. *If T is a symmetric pseudo-commutative accessible 2-monad on Cat, the 2-category of T-algebras and pseudo-maps of T-algebras has a pseudo-monoidal pseudo-closed structure induced by the pseudo-commutative structure of T, coherently with respect to the closed structure of Cat.*

Corollary 1. *The 2-category of small categories with countable products, countable product preserving functors, and natural transformations is pseudo-monoidal pseudo-closed, coherently with respect to the closed structure of Cat.*

The heart of this result as it applies to us is that the construction that sends a pair of small categories C and D with countable products to the category $CP(C, D)$ of countable product preserving functors from C to D is a well behaved construction. Moreover, for any small categories C and C' with countable products, there is a small category $C \otimes C'$ with countable products together with a well behaved equivalence of categories between $CP(C, FP(C, D))$ and $CP(C \otimes C', D)$ natural in D. The theorem only determines the construction $C \otimes C'$ up to coherent equivalence of categories, but, when restricted to countable Lawvere theories, it agrees up to equivalence with the construction we gave at the start of the section. Thus we may conclude the following:

Theorem 11. *The construction $L \otimes L'$ on countable Lawvere theories extends to a coherent pseudo-monoidal pseudo-closed structure on the 2-category of small categories with countable products, and, for any small category C with countable products, there is a coherent equivalence of categories between $Mod(L \otimes L', C)$ and $Mod(L, Mod(L', C))$.*