Combining Continuations with Other Effects [*]

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

P. B. Levy
School of Computer Science
University of Birmingham
Birmingham B15 2TT
England
P.B.Levy@cs.bham.ac.uk

G. Plotkin & J. Power
School of Informatics
University of Edinburgh
Edinburgh EH9 3JZ
Scotland
gdp@inf.ed.ac.uk, ajp@inf.ed.ac.uk

ABSTRACT

We consider the combination of continuations with other effects, starting from Eugenio Moggi's monadic framework. It turns out that the usual combinations can be analysed in one of three distinct ways: sum, tensor, and by applying the continuations monad transformer $C(-)$, and each such way has a natural universal characterisation. Mathematically this requires delicate considerations on large categories; conceptually it remains unclear why the combinations are computationally correct.

Categories and Subject Descriptors

F.3.2 [**Logics and Meanings of Programs**]: Semantics of Programming Languages; D.3.1 [**Programming Languages**]: Formal Definitions and Theory

General Terms

Languages, Theory

Keywords

Continuations, effects, monad, monad transformer

1. INTRODUCTION

A fundamental question, in modelling computational effects, is how to give a unified semantic account of modularity, i.e., a mathematical theory that supports the various combinations one naturally makes of computational effects such as exceptions, side-effects, interactive input/output, nondeterminism, and, particularly for this workshop, continuations [2, 3, 5]. We have begun to give such an account over recent years for all of these effects other than continuations [8], describing the *sum* and the *tensor*, or *commutative combination*, of effects, starting from Eugenio Moggi's proposal to use monads to give semantics for each individual effect [15]. That has yielded the most commonly used combinations of the various effects. Here, we extend our account to include continuations.

We consider three distinct ways in which continuations combine with the other effects: sum, tensor, and by applying the continuations monad transformer $C(-)$; we analyse each of these in the following three sections. We did not include continuations in [8] as they are of a different nature, both computationally and mathematically, to the other effects. Computationally, the other effects arise naturally from algebraic operations and equations [16], but continuations seem not to do so and seem better developed in terms of control operators such as Felleisen's $\mathcal{C}$ operator [4, 7, 6]. Mathematically, the monads generated by the other effects all have rank [1, 9, 10], which implies that the sum and tensor of any two of them always exist. The continuations monad R^{R^-} does not have rank; a sum with it does not exist in general; and nor might a tensor with it.

One might ask, why go to this trouble? After all, Moggi and his colleagues have already given us a notion of monad transformer [2, 3], and the constructions we develop here all yield known monad transformers. But the work on monad transformers has not explained how to derive a monad transformer from the associated monad; and nor has it shown how to extend operations to the combination of two effects. Our analysis here and in [8] yields such a derivation in the cases of sum and tensor, and it shows there is no fundamental asymmetry as the monad transformers are derived from the symmetric sum and tensor. It also allows us to redefine algebraic operations systematically: there are canonical monad maps from T to $T + T'$, to $T \otimes T'$, and to $C(T)$, and operations, qua *generic effects* [8], extend systematically along monad maps; this yields *operation transformers.*

For control by itself, given a cartesian category with a strong monad T and Kleisli exponentials, we assume a distributive initial object 0 (i.e., $x \times 0 \cong 0$) and a family $\mathcal{C}_x : \neg\neg x \longrightarrow Tx$, where $\neg x =_{\mathrm{def}} T0^x$, inverse to the canonical strong monad map $d_T : T \longrightarrow \neg\neg$, cf. [7, 6]. These axioms also hold in the models given below for the combination of control with all the other effects, apart from exceptions. There $\mathcal{C}$ is a retract, not an isomorphism; the counterexample in [12] shows that this also holds operationally.

We end the paper with a formula for a typical combination of effects, making clear the elegance and simplicity obtained by our analysis, followed by some discussion of what remains to be done. For simplicity of exposition, we generally restrict attention to monads on Set.

[*] This work has been done with the support of EPSRC grants GR/M56333 and GR/N64571.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$5.00.

2. SUM AND EXCEPTIONS

The sum of effects appears in the combination of exceptions with all other computational effects we consider, including continuations. It follows from Theorem 1 below that the sum of an arbitrary monad T with the (simple) exceptions monad $T_E =_{\mathrm{def}} - + E$ in the category of monads on *Set* is $T(- + E)$. So $R^{R^-} + T_E$ is $R^{R^{(-+E)}}$, the usual combination of the two effects [17].

On the other hand, the sum of R^{R^-} with even a very simple monad need not exist:

Example 1. Let T_u be the monad on *Set* generated by a single unary operation, i.e., $T_u = Id^*$. If $R^{R^-} + T_u$ did exist, then, by the analysis below, its value at $\emptyset$ would solve the isomorphism equation $Z \cong R^{R^{Z \times \mathbb{N}}}$. But that fails for evident cardinality reasons when $|R| > 1$.

So we cannot take a sum of the continuations monad with an arbitrary monad, but we can take a sum of the exceptions monad with an arbitrary monad. The argument follows.

First, for notation, given an endofunctor Σ on a category $\mathcal{A}$, if the forgetful functor from Σ-*alg* to $\mathcal{A}$ has a left adjoint, we say that the resulting monad is the *free* monad on Σ and write it as Σ^*. Explicitly, assuming $\mathcal{A}$ has binary sums, Σ^* is $\mu y.(\Sigma y + -)$, with one existing if and only if the other does (where, for any endofunctor F, we write $\mu y.Fy$ for the initial F-algebra, if it exists). In [8], we proved:

Theorem 1. *Let Σ be an endofunctor on a category with binary sums for which Σ^* exists, and let T be a monad. If $\mu y.T(\Sigma y + -)$ exists, then the sum of monads $T + \Sigma^*$ exists and is given by a canonical monad structure on the functor $\mu y.T(\Sigma y + -)$.*

Theorem 1 includes the example of exceptions, taking Σ to be the constantly E functor; one can also give a direct proof [8, 14]. Note that the sum of monads is not the sum of the underlying functors, i.e. is not given pointwise.

Although not stated in [8], there is a converse: if the sum exists, it must be given by the formula. The conclusion in Example 1 follows; a similar argument shows the sum of R^{R^-} with the I/O monad $T_{I/O} =_{\mathrm{def}} \mu Y.(Y^I + O \times Y + -)$ also fails to exist.

In regard to Felleisen's $\mathcal{C}$ operator, we have:

Proposition 1. *If d_T has an inverse, then d_{T+T_E} has a left inverse, but need not itself be invertible.*

3. TENSOR AND SIDE-EFFECTS

The natural combination of the continuations monad with the side-effects monad is $(R^S)^{(R^S)^-}$, as used in Scheme [11]. It follows from Theorem 2 below that this is the *tensor*, or *commutative combination*, of the two monads. The general notion of tensor is not easy to motivate in terms of monads: it exists more naturally as a construct on, e.g., *countable Lawvere theories*, an equivalent formulation of the notion of monad with countable rank [8].

There are two possible ways to extend the notion of tensor from countable Lawvere theories to arbitrary monads. One is to define a notion of a theory of arbitrary size, equivalent to arbitrary monads, and then generalise to such theories. The other is to translate the construction of a tensor product of countable Lawvere theories into monadic terms. Here, we do the latter:

Definition 1. Given two monads T and T', the monad $T \otimes T'$, which we call the *tensor product* of T and T' if it exists, is defined by the universal property of having monad maps α and α' from T and T' to $T'' = T \otimes T'$, subject to the commutativity of

$$\begin{tikzcd}
TX \times T'Y \arrow[r, "\alpha \times \alpha'"] \arrow[d, "\alpha \times \alpha'"'] & T''X \times T''Y \arrow[d, "\sigma"] \\
T''X \times T''Y \arrow[r, "\bar{\sigma}"'] & T''(X \times Y)
\end{tikzcd}$$

where σ and $\bar{\sigma}$ are the two canonical maps induced by the strength of T'' (which is uniquely determined for any monad on *Set*).

The coherence condition of the tensor product, expressed in terms of Lawvere theories, is the assertion that the operations of one theory commute with those of the other. There do not seem to be *computationally natural* operations and equations that generate the continuations monad, but the Lawvere theory formulation is more natural for most other examples [8].

In general, the tensor product of two arbitrary monads seems not to exist, but we do not know a counterexample. We do, however, have some partial positive results: the tensor product of 2^{2^-} with any monad with rank exists, and the tensor product of any continuations monad with any monad T whose Lawvere theory contains a constant is the trivial (= constantly 1) monad.

Theorem 2. *If T is an arbitrary monad, the tensor product of T with T_S exists and is given by the monad $T(S \times -)^S$.*

One can prove this theorem by brute force from the definition; a more elegant proof follows from the formulation of tensor product in terms of Lawvere theories applied to the characterisation of global state in [16].

In regard to Felleisen's $\mathcal{C}$ operator, we have the following result:

Proposition 2. *If d_T is invertible (has a left inverse) then $d_{T \otimes T_S}$ is invertible (has a left inverse).*

4. THE CONTINUATIONS MONAD TRANSFORMER

The continuations monad transformer, $C(T) =_{\mathrm{def}} TR^{TR^-}$ [3, 2], is a unary construction, whereas sum and tensor are binary constructions, applied to continuations and some other effect. It applies naturally to nondeterminism and to I/O, e.g., taking the finite non-empty powerset monad $\mathcal{F}^+$ to $(\mathcal{F}^+R)^{(\mathcal{F}^+R)^-}$. Note that for any monad T, the canonical map $d_{C(T)} : C(T) \longrightarrow \neg\neg$ is invertible.

The continuations monad transformer seems to be more primitive than the continuations monad: we naturally obtain the monad from the transformer, as $R^{R^-} = C(Id)$, but we do not see any principled way to derive $C(-)$ from R^{R^-}. Moreover, it seems there is, in general, no monad map from R^{R^-} to $C(T)$, whereas there is one from T to $C(T)$. Nevertheless, a universal characterisation is available:

THEOREM 3. *Given a monad T and a set R, there is a universal monad map $\alpha : T \longrightarrow C(T)$ together with a T-algebra isomorphism $f : T(R) \cong C(T)(0)$. The universal property says that given any such maps $\alpha' : T \longrightarrow S$ and $f' : T(R) \cong S(0)$, there is a unique $\beta : S \longrightarrow C(T)$ such that $\beta\alpha' = \alpha$ and $\beta_0 f' = f$.*

Observe in particular the directions involved with the universal property: there is some, but not full, reversal from that for sum and tensor. This property can also be described naturally in terms of Lawvere theories.

Finally, we note that the transformer is parametrised on R, but this can be extended to an arbitrary T-algebra (A, a), replacing TR by A [13]; Theorem 3 then extends too. Curiously, the combination of the state monad with the continuations monad, analysed above as a tensor, can also be regarded as being of this form, taking the T_S-algebra to be (R^S, a) with the evident $a : (S \times R^S)^S \to R^S$.

5. DISCUSSION

Using the constructs we have developed, we propose formulae for combining exceptions, side-effects, interactive input/output, (binary) nondeterminism and continuations. For all of them together we propose:

$$T_E + (T_S \otimes C(T_{I/O} + \mathcal{F}^+))$$

which we note is linear, having the form

$$M_E(M_S(C(M_{I/O}(\mathcal{F}^+))))$$

with each M derived from $+$ or $\otimes$ applied to a particular monad. To omit effects, omit the corresponding parts of the formula. We have no independent justification of these proposals, but they are consistent with all the cases we know.

The main thing missing in our account of the combinations of continuations with other effects is an understanding of *why* they are the right choices (if indeed they are!); one puzzle here is that the combination of state with continuations can be explained in two different ways. For the combination of effects other than continuations with each other, the choices were justified computationally, in terms of the equations involving the sets of operations inherited from each effect [8]; perhaps there is some analogous explanation involving the interaction between the operations and Felleisen's $\mathcal{C}$ operator.

A closely related question concerns finding the right axiomatisation of Moggi's computational λ-calculus with continuations and the various effects, cf. [12, 18]. One would also like to understand combinations of continuations with local effects, such as local store or exceptions.

6. REFERENCES

[1] J. Adámek & J. Rosický, *Locally Presentable and Accessible Categories*, London Mathematical Society Lecture Note Series, **189**, CUP, 1994.

[2] N. Benton, J. Hughes & E. Moggi, Monads and Effects, *Proc. APPSEM 2000*, LNCS **2395**, 42–122, 2002.

[3] P. Cenciarelli & E. Moggi, A Syntactic Approach to Modularity in Denotational Semantics, *Proc. 5th CTCS*, CWI Technical report, 1993.

[4] M. Felleisen, D. P. Friedman, et al, A Syntactic Theory of Sequential Control. *TCS* **52**, 205–237, 1987.

[5] M. P. Fiore, A. Jung, et al, Domains and Denotational Semantics: History, Accomplishments and Open Problems, *Bulletin of EATCS* (59), 227–256, 1996.

[6] C. Fuhrmann & H. Thielecke, On the Call-by-Value CPS Transform and its Semantics, *I. & C.*, to appear.

[7] M. Hofmann, Sound and Complete Axiomatisations of Call-by-Value Control Operators, *MSCS*, **5**(4), 461–482, 1995.

[8] J. M. E. Hyland, G. D. Plotkin & A. J. Power, Combining Effects: Sum and Tensor, submitted.

[9] G. M. Kelly, A Unified Treatment of Transfinite Constructions for Free Algebras, Free Monoids, Colimits, Associated Sheaves, and so on, *Bull. Austral. Math. Soc.*, **22**, 1–83, 1980.

[10] G. M. Kelly & A. J. Power, Adjunctions whose Counits are Coequalizers, and Presentations of Finitary Enriched Monads, *JPAA*, **89**, 163–179, 1993.

[11] R. Kelsey, W. Clinger & J. Rees (eds.), Revised^5 Report on the Algorithmic Language Scheme, *HOSC*, **11**, 7–105, 1998.

[12] J. Laird, Exceptions, Continuations and Macro-expressiveness, *Proc. ESOP*, LNCS **2305**, 133-146, 2002.

[13] P. B. Levy, *Call-By-Push-Value*, Ph.D. thesis, Queen Mary, University of London, 2001.

[14] C. Luth & N. Ghani, Composing Monads Using Coproducts, *Proc. ICFP '02*, 133–144, ACM Press, 2002.

[15] E. Moggi, Notions of Computation and Monads, *I. & C.*, **93**(1) 55–92, 1991.

[16] G. D. Plotkin & A. J. Power, Notions of Computation Determine Monads, *Proc. FOSSACS '02*, LNCS, **2303**, 342–356, 2002.

[17] H. Thielecke, Comparing Control Constructs by Double-Barrelled CPS, *HOSC*, **15**, 141–160, 2002.

[18] H. Thielecke, Contrasting Exceptions and Continuations, Submitted.