MFPS 17 Preliminary Version

Semantics for algebraic operations

Gordon Plotkin and John Power [1]

Laboratory for the Foundations of Computer Science
University of Edinburgh
King's Buildings, Edinburgh EH9 3JZ, SCOTLAND

Abstract

Given a category C with finite products and a strong monad T on C, we investigate axioms under which an ObC-indexed family of operations of the form $\alpha_x : (Tx)^n \longrightarrow Tx$ provides a definitive semantics for algebraic operations added to the computational λ-calculus. We recall a definition for which we have elsewhere given adequacy results for both big and small step operational semantics, and we show that it is equivalent to a range of other possible natural definitions of algebraic operation. We outline examples and non-examples and we show that our definition is equivalent to one for call-by-name languages with effects too.

1 Introduction

Eugenio Moggi, in [6,8], introduced the idea of giving a unified category theoretic semantics for computational effects such as nondeterminism, probabilistic nondeterminism, side-effects, and exceptions, by modelling each of them uniformly in the Kleisli category for an appropriate strong monad on a base category C with finite products. He supported that construction by developing the computational λ-calculus or λ_c-calculus, for which it provides a sound and complete class of models. The computational λ-calculus is essentially the same as the simply typed λ-calculus except for the essential fact of making a careful systematic distinction between computations and values. However, it does not contain operations, and operations are essential to any programming language. So here, in beginning to address that issue, we provide a unified semantics for algebraic operations, supported by equivalence theorems to indicate definitiveness of the axioms.

We distinguish here between algebraic operations and arbitrary operations. The former are, in a sense we shall make precise, a natural generalisation, from

[1] This work is supported by EPSRC grant GR/L89532: Notions of computability for general datatypes.

This is a preliminary version. The final version will be published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

Set to an arbitrary category C with finite products, of the usual operations of universal algebra. The key point is that the operations

$$\alpha_x : (Tx)^n \longrightarrow Tx$$

are parametrically natural in the Kleisli category for a strong monad T on C, as made precise in Definition 2.1: in that case, we say that the monad T supports the operations; the leading class of examples has T being generated by the operations subject to equations accompanying them. Examples of such operations are those for nondeterminism and probabilistic nondeterminism, and for raising exceptions. A non-example is given by an operation for handling exceptions.

In a companion paper [11], we have given the above definition, given a syntactic counterpart in terms of the computational λ-calculus, and proved adequacy results for small and big-step operational semantics. But such results alone leave some scope for a precise choice of appropriate semantic axioms. So in this paper, we prove a range of equivalence results, which we believe provide strong evidence for a specific choice of axioms, namely those for parametric naturality in the Kleisli category as mentioned above. Our most profound result is essentially about a generalisation of the correspondence between finitary monads and Lawvere theories from *Set* to a category with finite products C and a strong monad T on C: this result characterises algebraic operations as generic effects. The generality of our analysis is somewhat greater than in the study of enriched Lawvere theories in [12]: the latter require C to be locally finitely presentable as a closed category, which is not true of all our leading examples.

Moggi gave a semantic formulation of a notion of operation in [7], with an analysis based on his computational metalanguage, but he only required naturality of the operations in C, and we know of no way to provide operational semantics in such generality. Our various characterisation results do not seem to extend to such generality either. Evident further work is to consider how other operations such as those for handling exceptions should be modelled. That might involve going beyond monads, as Moggi has suggested to us; one possibility is in the direction of dyads [13].

We formulate our paper in terms of a strong monad T on a category with finite products C. We could equally formulate it in terms of closed *Freyd*-categories in the spirit of [1], which provides a leading example for us in its analysis of finite nondeterminism.

The paper is organised as follows. In Section 2, we recall the definition of algebraic operation given in [11] and we exhibit some simple reformulations of it. In Section 3, we give direct equivalent versions of these statements in terms of enrichment under the assumption that C is closed. In Section 4, we give a more substantial reformulation of the notion in terms of operations on homs, both when C is closed and more generally when C is not closed. In Section 5, we give what we regard as the most profound result of the paper, which is a formulation in terms of generic effects, generalising a study of Lawvere theories. Finally, in Section 6, we characterise algebraic operations in terms of operations on the category T-Alg, as this gives an indication of how to incorporate call-by-name languages with computational effects into the picture. And we give conclusions and an outline of possible future directions in Section 7.

2 Algebraic operations and simple equivalents

In this section, we give the definition of algebraic operation as we made it in [11]. In that paper, we gave the definition and a syntactic counterpart in terms of the computational λ-calculus, and we proved adequacy results for small and big-step operational semantics for the latter in terms of the former. Those results did not isolate definitive axioms for the notion of algebraic operation. So in this section, we start with a few straightforward equivalence results on which we shall build later.

We assume we have a category C with finite products together with a strong monad $< T, \eta, \mu, st >$ on C with Kleisli exponentials, i.e., such that for all objects x and z of C, the functor $C_T(-\times x, z) : C^{op} \longrightarrow Set$ is representable. We do not take C to be closed in general: we shall need to assume it for some later results, but we specifically do not want to assume it in general, and we do not require it for any of the results of this section.

Given a map $f : y \times x \longrightarrow Tz$ in C, we denote the parametrised lifting of f, i.e., the composite

$$y \times Tx \xrightarrow{st} T(y \times x) \xrightarrow{Tf} T^2 z \xrightarrow{\mu_z} Tz$$

by $f^\dagger : y \times Tx \longrightarrow Tz$.

Definition 2.1 An *algebraic operation* is an ObC-indexed family of maps

$$\alpha_x : (Tx)^n \longrightarrow Tx$$

such that for every map $f : y \times x \longrightarrow Tz$ in C, the diagram

$$\begin{tikzcd}
y \times (Tx)^n \arrow[r, "\langle f^\dagger \cdot (y \times \pi_i)\rangle_{i=1}^n"] \arrow[d, "y \times \alpha_x"'] & (Tz)^n \arrow[d, "\alpha_z"] \\
y \times Tx \arrow[r, "f^\dagger"'] & Tz
\end{tikzcd}$$

commutes.

For some examples of algebraic operations, for $C = Set$, let T be the nonempty finite power-set monad with binary choice operations [9,1]; alternatively, let T be the monad for probabilistic nondeterminism with probabilistic choice operations [2,3]; or take T to be the monad for printing with printing operations [10]. Observe the non-commutativity in the latter example. One can, of course, generalise from Set to categories such as that of ω-cpo's, for instance considering the various power-domains together with binary choice operators. One can also consider combinations of these, for instance to model internal and external choice operations. Several of these examples are treated in detail in [11].

There are several equivalent formulations of the coherence condition of the definition. Decomposing it in a maximal way, we have

Proposition 2.2 *An ObC-indexed family of maps*

$$\alpha_x : (Tx)^n \longrightarrow Tx$$

is an algebraic operation if and only if

(i) *α is natural in C*

(ii) *α respects st in the sense that*

$$\begin{tikzcd}
y \times (Tx)^n \arrow[r, "\langle st \cdot (y \times \pi_i) \rangle_{i=1}^n"] \arrow[d, "y \times \alpha_x"'] & (T(y \times x))^n \arrow[d, "\alpha_{y \times x}"] \\
y \times Tx \arrow[r, "st"'] & T(y \times x)
\end{tikzcd}$$

commutes

(iii) *α respects μ in the sense that*

$$\begin{tikzcd}
(T^2x)^n \arrow[r, "\mu_x^n"] \arrow[d, "\alpha_{Tx}"'] & (Tx)^n \arrow[d, "\alpha_x"] \\
T^2x \arrow[r, "\mu_x"'] & Tx
\end{tikzcd}$$

commutes.

Proof. It is immediately clear from our formulation of the definition and the proposition that the conditions of the proposition imply the coherence requirement of the definition. For the converse, to prove naturality in C, put $y = 1$ and, given a map $g : x \longrightarrow z$ in C, compose it with η_z and apply the coherence condition of the definition. For coherence with respect to st, take $f : y \times x \longrightarrow Tz$ to be $\eta_{y \times x}$. And for coherence with respect to μ, put $y = 1$ and take f to be id_{Tx}. $\square$

There are other interesting decompositions of the coherence condition of the definition too. In the above, we have taken T to be an endo-functor on C. But one often also writes T for the right adjoint to the canonical functor $J : C \longrightarrow C_T$ as the behaviour of the right adjoint on objects is given precisely by the behaviour of T on objects. So with this overloading of notation, we have functors $(T-)^n : C_T \longrightarrow C$ and $T : C_T \longrightarrow C$, we can speak of natural transformations between them, and we have the following proposition.

Proposition 2.3 *An ObC-indexed family of maps*

$$\alpha_x : (Tx)^n \longrightarrow Tx$$

is an algebraic operation if and only if α is natural in C_T and α respects st.

In another direction, as we shall investigate further below, it is sometimes convenient to separate the μ part of the coherence condition from the rest of it. We can do that with the following somewhat technical result.

Proposition 2.4 *An ObC-indexed family*

$$\alpha_x : (Tx)^n \longrightarrow Tx$$

forms an algebraic operation if and only if α respects μ and, for every map $f : y \times x \longrightarrow z$ in C, the diagram

$$\begin{tikzcd}
y \times (Tx)^n \arrow[r, "\langle st \cdot (y \times \pi_i) \rangle_{i=1}^n"] \arrow[d, "y \times \alpha_x"'] & (T(y \times x))^n \arrow[r, "(Tf)^n"] & (Tz)^n \arrow[d, "\alpha_z"] \\
y \times Tx \arrow[r, "st"'] & T(y \times x) \arrow[r, "Tf"'] & Tz
\end{tikzcd}$$

commutes.

3 Equivalent formulations if C is closed

For our more profound results, it seems best first to assume that C is closed, explain the results in those terms, and later to drop the closedness condition and explain how to reformulate the results without essential change. So for the results in this section, we shall assume C is closed.

Let the closed structure of C be denoted by $[-,-]$. Given a monad $< T, \eta, \mu >$ on C, to give a strength for T is equivalent to giving an enrichment of T in C: given a strength, one has an enrichment

$$T_{x,y} : [x, y] \longrightarrow [Tx, Ty]$$

given by the transpose of

$$[x, y] \times Tx \xrightarrow{st} T([x, y] \times x) \xrightarrow{Tev} Ty$$

and given an enrichment of T, one has a strength given by the transpose of

$$x \longrightarrow [y, x \times y] \xrightarrow{T_{y,x\times y}} [Ty, T(x \times y)]$$

It is routine to verify that the axioms for a strength are equivalent to the axioms for an enrichment. So, given a strong monad $< T, \eta, \mu, st >$ on C, the monad T is enriched in C, and so is the functor $(-)^n : C \longrightarrow C$.

The category C_T also canonically acquires an enrichment in C, i.e, the homset $C_T(x, y)$ of C_T lifts to a homobject of C: the object $[x, Ty]$ of C acts as a homobject, applying the functor $C(1, -) : C \longrightarrow Set$ to it giving the homset $C_T(x, y)$; composition

$$C_T(y, z) \times C_T(x, y) \longrightarrow C_T(x, z)$$

lifts to a map in C

$$[y, Tz] \times [x, Ty] \longrightarrow [x, Tz]$$

determined by taking a transpose and applying evaluation maps twice and each of the strength and the multiplication once; and identities and the axioms for a category lift too.

The canonical functor $J : C \longrightarrow C_T$ becomes a C-enriched functor with a C-enriched right adjoint. The main advantage of the closedness condition for us is that it allows us to dispense with the parametrisation of the naturality, or equivalently with the coherence with respect to the strength, as follows.

Proposition 3.1 *If C is closed, an ObC-indexed family*

$$\alpha_x : (Tx)^n \longrightarrow Tx$$

forms an algebraic operation if and only if

$$\begin{tikzcd}
{[x, Tz]} \arrow[r, "(-)^n \cdot {[Tx, \mu_z]} \cdot T_{x,Tz}"] \arrow[d, "{[Tx, \mu_z]} \cdot T_{x,Tz}"'] & {[(Tx)^n, (Tz)^n]} \arrow[d, "{[(Tx)^n, \alpha_z]}"] \\
{[Tx, Tz]} \arrow[r, "{[\alpha_x, Tz]}"'] & {[(Tx)^n, Tz]}
\end{tikzcd}$$

commutes.

The left-hand vertical map in the diagram here is exactly the behaviour of the C-enriched right adjoint $T : C_T \longrightarrow C$ to the canonical C-enriched functor $J : C \longrightarrow C_T$ on homs, and the top horizontal map is exactly the behaviour of the C-enriched functor $(T-)^n : C_T \longrightarrow C$ on homs. So the coherence condition in the proposition is precisely the statement that α forms a C-enriched natural transformation from the C-enriched functor $(T-)^n : C_T \longrightarrow C$ to the C-enriched functor $T : C_T \longrightarrow C$.

Proof. Given a map $f : y \times x \longrightarrow Tz$ in C, the transpose of the map gives a map from y to $[x, Tz]$. Precomposing the coherence condition here with that map, then transposing both sides, one obtains the coherence condition of the definition. For the converse, given a map $g : y \longrightarrow [x, Tz]$, taking its transpose, using the coherence condition of the definition, and transposing back again, shows that the above square precomposed with g commutes. So by the Yoneda lemma, we are done. $\square$

The same argument can be used to give a further characterisation of the notion of algebraic operation if C is closed by modifying Proposition 2.4. This yields

Proposition 3.2 *If C is closed, an ObC-indexed family*

$$\alpha_x : (Tx)^n \longrightarrow Tx$$

forms an algebraic operation if and only if α respects μ and

$$\begin{tikzcd}
{[x,z]} \arrow[r, "(-)^n \cdot T_{x,z}"] \arrow[d, "T_{x,z}"'] & {[(Tx)^n, (Tz)^n]} \arrow[d, "{[(Tx)^n, \alpha_z]}"] \\
{[Tx, Tz]} \arrow[r, "{[\alpha_x, Tz]}"'] & {[(Tx)^n, Tz]}
\end{tikzcd}$$

commutes.

This proposition says that if C is closed, an algebraic operation is exactly a C-enriched natural transformation from the C-enriched functor $(T-)^n : C \longrightarrow C$ to the C-enriched functor $T : C \longrightarrow C$ that is coherent with respect to μ.

4 Algebraic operations as operations on homs

In our various formulations of the notion of algebraic operation so far, we have always had an ObC-indexed family

$$\alpha_x : (Tx)^n \longrightarrow Tx$$

and considered equivalent conditions on it under which it might be called an algebraic operation. In computing, this amounts to considering an operator on expressions. But there is another approach in which arrows of the category C_T may be seen as primitive, regarding them as programs. This was the underlying idea of the reformulation [1] of the semantics for finite nondeterminism of [9]. So we should like to reformulate the notion of algebraic operation in these terms. Proposition 3.1 allows us to do that. In order to explain the reason for the coherence conditions, we shall start by expressing the result assuming C is closed; after which we shall drop the closedness assumption and see how the result can be re-expressed using parametrised naturality.

We first need to explain an enriched version of the Yoneda lemma as in [4]. If D is a small C-enriched category, then D^{op} may also be seen as a C-enriched category. We do not assume C is complete here, but if we did, then we would have a C-enriched functor category $[D^{op}, C]$ and a C-enriched Yoneda embedding

$$Y_D : D \longrightarrow [D^{op}, C]$$

The C-enriched Yoneda embedding Y_D is a C-enriched functor and it is fully faithful in the strong sense that the map

$$D(x, y) \longrightarrow [D^{op}, C](D(-, x), D(-, y))$$

is an isomorphism in the category C: see [4] for all the details. It follows by applying the functor $C(1, -) : C \longrightarrow Set$ that this induces a bijection from the set of maps from x to y in D to the set of C-enriched natural transformations from the C-enriched functor $D(-, x) : D^{op} \longrightarrow C$ to the C-enriched functor $D(-, y) : D^{op} \longrightarrow C$.

This is the result we need, except that we do not want to assume that C is complete, and the C-enriched categories of interest to us are of the form C_T, so in general are not small. These are not major problems although they go a little beyond the scope of the standard formulation of enriched category theory in [4]: one can embed C into a larger universe C' just as one can embed Set into a larger universe Set' when necessary, and the required mathematics for the enriched analysis appears in [4]. We still have what can reasonably be called a Yoneda embedding of D into $[D^{op}, C]$, with both categories regarded as C'-enriched rather than C-enriched, and it is still fully faithful as a C'-enriched functor. However, we can formulate the result we need more directly without reference to C' simply by stating a restricted form of the enriched Yoneda lemma: letting $Fun_C(D^{op}, C)$ denote the (possibly large) category of C-enriched functors from D^{op} to C, the underlying ordinary functor

$$D \longrightarrow Fun_C(D^{op}, C)$$

of the Yoneda embedding is fully faithful.

We use this latter statement both here and in the following section. Now for our main result of this section under the assumption that C is closed.

Theorem 4.1 *If C is closed, to give an algebraic operation is equivalent to giving an $ObC^{op} \times ObC$ family of maps*

$$a_{y,x} : [y, Tx]^n \longrightarrow [y, Tx]$$

that is C-natural in y as an object of C^{op} and C-natural in x as an object of

C_T, i.e., such that

$$\begin{tikzcd}
{[y,Tx]^n \times [y',y]} \arrow[r, "\langle comp\cdot(\pi_i \times [y',y])\rangle_{i=1}^n"] \arrow[d, "a_{y,x}\times[y',y]"'] & {[y',Tx]^n} \arrow[d, "a_{y',x}"] \\
{[y,Tx]\times[y',y]} \arrow[r, "comp"'] & {[y',Tx]}
\end{tikzcd}$$

and

$$\begin{tikzcd}
{[x,Tz]\times[y,Tx]^n} \arrow[r, "\langle comp_K\cdot([x,Tz]\times\pi_i)\rangle_{i=1}^n"] \arrow[d, "{[x,Tz]\times a_{y,x}}"'] & {[y,Tz]^n} \arrow[d, "a_{y,z}"] \\
{[x,Tz]\times[y,Tx]} \arrow[r, "comp_K"'] & {[y,Tz]}
\end{tikzcd}$$

commute, where comp is the C-enriched composition of C and $comp_K$ is C-enriched Kleisli composition.

Proof. First observe that $[y,Tx]^n$ is isomorphic to $[y,(Tx)^n]$. Now, it follows from our C-enriched version of the Yoneda lemma that to give the data together with the first axiom of the proposition is equivalent to giving an ObC-indexed family

$$\alpha : (Tx)^n \longrightarrow Tx$$

By a further application of our C-enriched version of the Yoneda lemma, it follows that the second condition of the proposition is equivalent to the coherence condition of Proposition 3.1. □

As mentioned earlier, we can still state essentially this result even without the condition that C be closed. There are two reasons for this. First, for the paper, we have assumed the existence of Kleisli exponentials, as are essential in order to model λ-terms. But most of the examples of the closed structure of C we have used above are of the form $[y,Tx]$, which could equally be expressed as the Kleisli exponential $y \Rightarrow x$. The Kleisli exponential routinely extends to a functor

$$- \Rightarrow - : C_T^{op} \times C_T \longrightarrow C$$

Second, in the above, we made one use of a construct of the form $[y',y]$ with no T protecting the second object. But we can replace that by using the ordinary Yoneda lemma to express the first condition of the theorem in terms of maps $f : w \times y' \longrightarrow y$.

Summarising, we have

Corollary 4.2 *To give an algebraic operation is equivalent to giving an $ObC^{op} \times ObC$ family of maps*

$$a_{y,x} : (y \Rightarrow x)^n \longrightarrow (y \Rightarrow x)$$

in C, such that for every map $f : w \times y' \longrightarrow y$ in C, the diagram

$$\begin{tikzcd}
(y \Rightarrow x)^n \times w \times y' \arrow[r, "(f \Rightarrow x)^n \times w \times y'"] \arrow[d, "a_{y,x} \times f"'] & ((w \times y') \Rightarrow x)^n \times w \times y' \arrow[d, "ev \cdot (a_{w \times y', x} \times w \times y')"] \\
(y \Rightarrow x) \times y \arrow[r, "ev"'] & x
\end{tikzcd}$$

commutes, and the diagram

$$\begin{tikzcd}
(x \Rightarrow z) \times (y \Rightarrow x)^n \arrow[r, "\langle comp_K \cdot ((x \Rightarrow z) \times \pi_i)\rangle_{i=1}^n"] \arrow[d, "(x \Rightarrow z) \times a_{y,x}"'] & (y \Rightarrow z)^n \arrow[d, "a_{y,z}"] \\
(x \Rightarrow z) \times (y \Rightarrow x) \arrow[r, "comp_K"'] & (y \Rightarrow z)
\end{tikzcd}$$

commutes, where $comp_K$ is the canonical internalisation of Kleisli composition.

5 Algebraic operations as generic effects

In this section, we apply our formulation of the C-enriched Yoneda lemma to characterise algebraic operations in entirely different terms again as maps in C_T, i.e., in terms of generic effects. Observe that if C has an n-fold coproduct $\mathbf{n}$ of 1, the functor $(T-)^n : C_T \longrightarrow C$ is isomorphic to the functor $\mathbf{n} \Rightarrow - : C_T \longrightarrow C$. If C is closed, the functor $\mathbf{n} \Rightarrow -$ enriches canonically to a C-enriched functor, and that C-enriched functor is precisely the representable C-functor $C_T(\mathbf{n}, -) : C_T \longrightarrow C$, where C_T is regarded as a C-enriched category. So by Proposition 3.1 together with our C-enriched version of the Yoneda lemma, we immediately have

Theorem 5.1 *If C is closed, the C-enriched Yoneda embedding induces a bijection between maps $1 \longrightarrow \mathbf{n}$ in C_T and algebraic operations*

$$\alpha_x : (Tx)^n \longrightarrow Tx$$

This result is essentially just an instance of an enriched version of the identification of maps in a Lawvere theory with operations of the Lawvere theory. Observe that it follows that there is no mathematical reason to restrict attention to algebraic operations of arity n for a natural number n. We could just as well speak, in this setting, of algebraic operations of the form

$$\alpha_x : (\mathbf{a} \Rightarrow -) \longrightarrow (\mathbf{b} \Rightarrow -)$$

for any objects $\mathbf{a}$ and $\mathbf{b}$ of C. So for instance, we could include an account of infinitary operations as one might use to model operations involved with state. For specific choices of C such as $C = Poset$, one could consider more exotic arities such as that given by Sierpinski space.

Once again, by use of parametrisation, we can avoid the closedness assumption on C here, yielding the stronger statement

Theorem 5.2 *Functoriality of* $- \Rightarrow - : C_T^{op} \times C_T \longrightarrow C$ *in its first variable induces a bijection from the set of maps* $1 \longrightarrow \mathbf{n}$ *in* C_T *to the set of algebraic operations*

$$\alpha_x : (Tx)^n \longrightarrow Tx$$

We regard this as the most profound result of the paper. This result shows that to give an algebraic operation is equivalent to giving a generic effect, i.e., a constant of type the arity of the operation. For example, to give a binary nondeterministic operator for a strong monad T is equivalent to giving a constant of type 2, and to give equations to accompany the operator is equivalent to giving equations to be satisfied by the constant. The leading example here has T being the non-empty finite powerset monad or a powerdomain. Given a nondeterministic operator $\vee$, the constant is given by $true \vee false$, and given a constant c, the operator is given by $M \vee N =$ if c then M else N. There are precisely three non-empty finite subsets of the two element set, and accordingly, there are precisely three algebraic operations on the non-empty finite powerset monad, and they are given by the two projections and choice.

The connection of this result with enriched Lawvere theories [12] is as follows. If C is locally finitely presentable as a closed category, one can define a notion of finitary C-enriched monad on C and a notion of C-enriched Lawvere theory, and prove that the two are equivalent, generalising the usual equivalence in the case that $C = Set$. Given a finitary C-enriched monad T, the corresponding C-enriched Lawvere theory is given by the full sub-C-category of C_T determined by the finitely presentable objects. These include all finite coproducts of 1. So our results here exactly relate maps in the Lawvere theory with algebraic operations, generalising Lawvere's original idea. Of course, in this paper, we do not assume the finiteness assumptions on either the category C or the monad T, but our result here is essentially the same.

Theorem 5.2 extends with little fuss to the situation of finitely presentable objects $\mathbf{a}$ and $\mathbf{b}$; one just requires a suitable refinement of the construct $(T-)^n$ to account for $\mathbf{a}$ and $\mathbf{b}$ being objects of C rather than finite numbers. This follows readily by inspection of the work of [12], and, in a special case, it seems to provide an account of some of the operations associated with state, as suggested to us by Moggi.

6 Algebraic operations and the category of algebras

Finally, in this section, we characterise the notion of algebraic operation in terms of the category of algebras $T\text{-}Alg$. The co-Kleisli category of the comonad on $T\text{-}Alg$ induced by the monad T is used to model call-by-name languages with effects, so this formulation gives us an indication of how to generalise our analysis to call-by-name computation or perhaps to some combination of call-by-value and call-by-name, cf [5].

If C is closed and has equalisers, generalising Lawvere, the results of the previous section can equally be formulated as equivalences between algebraic operations and operations

$$\alpha_{(A,a)} : U(A,a)^n \longrightarrow U(A,a)$$

natural in (A,a), where $U : T\text{-}Alg \longrightarrow C$ is the C-enriched forgetful functor: equalisers are needed in C in order to give an enrichment of $T\text{-}Alg$ in C. We prove the result by use of our C-enriched version of the Yoneda lemma again, together with the observation that the canonical C-enriched functor $I : C_T \longrightarrow T\text{-}Alg$ is fully faithful. Formally, the result is

Theorem 6.1 *If C is closed and has equalisers, the C-enriched Yoneda embedding induces a bijection between maps $1 \longrightarrow \mathbf{n}$ in C_T and C-enriched natural transformations*

$$\alpha : (U-)^n \longrightarrow U- \, .$$

Combining this with Theorem 5.1, we have

Corollary 6.2 *If C is closed and has equalisers, to give an algebraic operation*

$$\alpha_x : (Tx)^n \longrightarrow Tx$$

is equivalent to giving a C-enriched natural transformation

$$\alpha : (U-)^n \longrightarrow U.$$

One can also give a parametrised version of this result if C is neither closed nor complete along the lines for C_T as in the previous section. It yields

Theorem 6.3 *To give an algeberaic operation*

$$\alpha_x : (Tx)^n \longrightarrow Tx$$

is equivalent to giving an $Ob(T\text{-}Alg)$-indexed family of maps

$$\alpha_{(A,a)} : U(A,a)^n \longrightarrow U(A,a)$$

such that, for each map

$$f : x \times U(A,a) \longrightarrow U(B,b)$$

commutativity of

$$\begin{tikzcd}
x \times TA \arrow[r, "x \times Tf"] \arrow[d, "x \times a"'] & x \times TB \arrow[d, "x \times b"] \\
x \times A \arrow[r, "x \times f"'] & x \times B
\end{tikzcd}$$

implies commutativity of

$$\begin{tikzcd}
x \times U(A,a)^n \arrow[r, "\langle f \cdot (x \times \pi_i) \rangle_{i=1}^n"] \arrow[d, "x \times \alpha_{(A,a)}"'] & U(B,b)^n \arrow[d, "\alpha_{(B,b)}"] \\
x \times U(A,a) \arrow[r, "f"'] & U(B,b)
\end{tikzcd}$$

7 Conclusions and Further Work

For some final comments, we note that little attention has been paid in the literature to the parametrised naturality condition on the notion of algebraic operation that we have used heavily here. And none of the main results of [11] used it, although they did require naturality in C_T. So it is natural to ask why that is the case.

For the latter point, in [11], we addressed ourselves almost exclusively to closed terms, and that meant that parametrised naturality of algebraic operations did not arise as we did not have any parameter.

Regarding why parametrised naturality does not seem to have been addressed much in the past, observe that for $C = Set$, every monad has a unique strength, so parametrised naturality of α is equivalent to ordinary naturality of α. More generally, if the functor $C(1,-) : C \longrightarrow Set$ is faithful, i.e., if 1 is a generator in C, then parametrised naturality is again equivalent to ordinary naturality of α. That is true for categories such as *Poset* and that of ω-cpo's, which have been the leading examples of categories studied in this regard. The reason we have a distinction is because we have not assumed that 1 is a generator, allowing us to include examples such as toposes or *Cat* for example.

Of course, in future, we hope to address other operations that are not algebraic, such as one for handling exceptions. It seems unlikely that the approach of this paper extends directly. Eugenio Moggi has recommended we look beyond monads. We should also like to extend and integrate this work with work addressing other aspects of giving a unified account of computational effects. We note here especially Paul Levy's work [5] which can be used to give accounts of both call-by-value and call-by-name in the same setting, and work on modularity [13], which might also help with other computational effects.

References

[1] Anderson, S.O., and A. J. Power, *A Representable Approach to Finite Nondeterminism*, Theoret. Comput. Sci. **177** (1997) 3–25.

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

[3] Jones, C., and G. D. Plotkin, *A Probabilistic Powerdomain of Evaluations*, Proc. LICS **4** (1989) 186–195.

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

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

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

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

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

[9] Plotkin, G.D., *A Powerdomain Construction*, SIAM J. Comput. **5** (1976) 452–487.

[10] Plotkin, G.D., "Domains," (http://www.dcs.ed.ac.uk/home/gdp/), 1983.

[11] Plotkin, G.D., and A. J. Power, *Adequacy for Algebraic Effects*, Proc. FOSSACS **2001** (to appear).

[12] Power, A.J., *Enriched Lawvere Theories*, Theory and Applications of Categories (2000) 83–93.

[13] Power, A.J., and E. P. Robinson, *Modularity and Dyads*, "Proc. MFPS **15**" Electronic Notes in Thoeret. Comp. Sci. **20**, 1999.