Math. Struct. in Comp. Science (1993), *vol.* 11, *pp.* 1–000

Premonoidal categories and notions of computation

JOHN POWER[1†] and EDMUND ROBINSON[2‡]

[1] *Department of Computer Science, University of Edinburgh, King's Buildings, Edinburgh EH9 3JZ, SCOTLAND.*
[2] *Department of Computer Science, Queen Mary and Westfield College, University of London, Mile End Road, London E14NS, UK.*

Received

We introduce the notions of premonoidal category and premonoidal functor, and show how these can be used in the denotational semantics of programming languages. We characterize the semantic definitions of Eugenio Moggi's monads as notions of computation, exhibit a representation theorem for our premonoidal setting in terms of monads, and give a fibrational setting for the structure.

1. Introduction

Eugenio Moggi, in (Moggi 1991), advocated the use of monads, equivalently Kleisli triples, to model what he called notions of computation. The idea was that one has a base category C such as the category of ω-cpos, in which to model ordinary, total programs, without any exotic features such as partiality, nondeterminism, side-effects, or exceptions. One considers a monad T on C and models the language extended by the specific notion of computation at hand in the Kleisli category for the monad. He also considered added conditions and structure on such a monad, namely the mono requirement, a tensorial strength (assuming C has finite products), and the commutativity of such strength.

Here, we reformulate his theory. We take the base category and the category providing the denotational semantics of the extended language as primitive, and add extra structure and conditions to those and the inclusion functor of the first into the second. This more flexible and somewhat more general framework allows us to model sequential composition of programs directly by sequential composition in our extended category, rather than by a sometimes complex construction involving a monad. It fits at least as well with all the examples, and better with some of them, such as side-effects. In particular, it allows us to characterize Milner *et.al.*'s control structures, (Mifsud *et al.* 1995; Power

† The first author acknowledges the support of ESPRIT Basic Research Action 6453: Types for proofs and programs and EPSRC grant GR/J84205: Frameworks for programming language semantics and logic.
‡ The second author acknowledges the support of ESPRIT Basic Research Action 6811: Categorical Logic in Computer Science II. This paper was written with the aid of an EPSRC Advanced Research Fellowship.

1996), for modelling concurrency. Also, our mathematical primitives correspond naturally and directly to computational phenomena such as product types and conditions upon them (see (Power and Robinson 1995)). Finally, it puts us in a more flexible position to model modularity in considering several notions of computation at once (see (Power and Robinson 1995) for an example).

In undertaking this study, we need to introduce a new concept, which we have called a premonoidal structure. In studying semantics of programming languages, categories with structure such as finite sums and products, exponentials, natural numbers objects, and directed colimits are used to model primitive programming constructions: premonoidal structure is used in a similar way. A premonoidal category is essentially a monoidal category except that the tensor need only be a functor of two variables and not necessarily be bifunctorial, i.e., given maps $f : x \longrightarrow y$ and $f' : x' \longrightarrow y'$, the evident two maps from $x \otimes x'$ to $y \otimes y'$ may differ.

For an instance of the use of premonoidal categories, in analysing side-effects, one may model a program from A to B by a function from $[A] \otimes S$ to $[B] \otimes S$, where S is the set of states and $[A]$ and $[B]$ denote types of values. Given another program, to be modelled by a function from $[A'] \otimes S$ to $[B'] \otimes S$, one obtains two different functions from $[A] \otimes [A'] \otimes S$ to $[B] \otimes [B'] \otimes S$, either of which could model the composite of the programs, dependent upon the order in which they are performed. One cannot describe such behaviour in a monoidal category of denotations of types and programs owing to the presence of bifunctoriality in the monoidal operation (see Example 3.4 for more detail). A similar situation arises in modelling the raising of exceptions and again in modelling nondeterminism by trees. For an analysis of the latter, see (Anderson and Power 1996). So we drop the assumption of bifunctoriality, giving rise to our definition, which appears in Section 3.

As for monoidal categories, a coherence question for premonoidal categories immediately arises: one wants all diagrams determined by the structural natural transformations to commute, giving unambiguous functions from a domain formed by an arbitrary number of tensor products. This allows us to speak unambiguously of the semantics of programs of several variables, rather than just two, as it in effect asserts that variable introduction be associative. So we take care with our definition in order to prove such a result in Section 3.

In Sections 2 and 3, we develop the definition and basic results for premonoidal categories. We address premonoidal structure in the presence of finite sums to model conditionals, and, with an eye to recursion, we ensure that our constructions enrich over the category of ω-cpos with least element, although not explicitly making much mention of it. We do not yet have a satisfactory account of higher order types in this generality. We do not imply that higher order types are unimportant: we just do not know how to handle them yet. There are several possible definitions of closed premonoidal category; but it is not clear yet which is definitive, as we have little relevant theory and no properly worked out examples. So we defer it.

In order to model exotic features of programming languages, it is useful to keep account of a category in which to model standard features. For instance, to model partiality, one analyses a category D of partial maps in terms of a category C of total maps. See, for instance, (Fiore 1996; Fiore and Plotkin 1994) for an account of partiality treated as a special case of the general approach we will take here. There is a structure preserving faithful functor $j : C \longrightarrow D$ that is the identity on objects, exhibiting the total maps as being among the partial maps and representing the idea that strictly terminating programs are among all programs, consistent with sequential composition. One is interested in finite products in C as they model tupling, or the modelling of programs of several variables. These are sent by j to a monoidal structure on D: but there may be many monoidal structures on D, and this one is typically not the finite product structure; so it is conveniently described as that monoidal structure on D that extends finite products on C. This situation is typical. For example, details for the modelling of nondeterminism appear in (Anderson and Power 1996). So it seems natural to develop machinery to handle structure on two categories, C and D, the latter in terms of the former. One wants the functor $j : C \longrightarrow D$ to preserve some of the structure of C in order to ensure that one preserves the modularity of the semantics of the basic programs when passing from the category of total maps to that of partial maps. In particular, one would like j to preserve finite sums, as that yields the preservation of the semantics of conditionals. One wants $j : C \longrightarrow D$ to be faithful as that ensures that one makes two basic programs equivalent in D if and only if they were made equivalent in C.

As we mentioned at the start, we are addressing the same question as that addressed by Moggi using monads. Our appoach is somewhat more general. In Section 4, we characterize all the main category theoretic definitions associated with Moggi's monads as an internalization of our pair of categories and functor with structure as above, and we give an example arising from concurrency of the greater generality of our approach. A priori, our external constructions strictly generalise his internal constructions. However, that generality is a little deceptive. In Sections 5 and 6, we prove a representation theorem to the effect that, subject to a size condition and a few side conditions, our construction $j : C \longrightarrow D$ can be embedded into the left adjoint part of a Kleisli construction, respecting the structure that internalizes to Moggi's structure on monads. We require commutativity of the monad, equivalently that the premonoidal structure on D be monoidal, to do that; but modulo that and the size and side conditions, our mathematical structures are formally equivalent to his. Finally, in Section 7, we show that premonoidal structure allows the construction of the simple fibration, as is usually applied to categories with finite products, and then we give a fibrational account of our leading result in those terms.

2. The other symmetric monoidal closed structure on Cat

It is well known that *Cat* is a cartesian closed category. It is not so well known that there is precisely one other symmetric monoidal closed structure on *Cat*. The exponential $C \longrightarrow D$ is given by the set of functors from C to D, with a morphism from g to h being the assignment of an arrow $\alpha_x : gx \longrightarrow hx$ to each object x of C. The composition is obvious. We shall call an arrow of $C \longrightarrow D$ a *transformation*. So, a natural transformation is a transformation subject to a commutativity condition.

The tensor product may be described in terms of a universal property: it is the universal

A for which one has, for each object x of C, a functor $h_x : D \longrightarrow A$ and for each object y of D, a functor $k_y : C \longrightarrow A$ such that $h_x y = k_y x$ for each (x, y). The unit of the tensor product is the unit category.

Explicitly, the tensor product $C \otimes D$ of C and D has as object set $ObC \times ObD$, and an arrow from (x, y) to (x', y') consists of a finite sequence of nonidentity arrows, with alternate arrows forming a directed path in C, and the others forming a directed path in D. Composition is given by concatenation, then cancellation accorded by the composition of C and D. The symmetry is obvious.

We shall denote this symmetric monoidal closed structure by Cat'. Observe that the identity is part of a monoidal functor from Cat' to Cat, the monoidal category determined by the cartesian structure of Cat.

We remark that $C \otimes - : Cat \longrightarrow Cat$ is not a 2-functor. An analysis of 2-categorical issues is unnecessary for our specific purposes here, so we defer it.

3. Premonoidal categories

Definition 3.1. A *strict premonoidal category* is a monoid in Cat'.

Every strict monoidal category is a strict premonoidal category, as follows from the fact that the identity is a monoidal functor from Cat' to Cat. A strict premonoidal category is essentially the same as a strict monoidal category except that one does not assert that the evident two maps from $x \otimes x'$ to $y \otimes y'$, given maps from x to y, and from x' to y', agree.

Just as monoidal categories are more common and usually more natural than strict monoidal categories, the leading examples of the phenomenon we want to address form premonoidal categories rather than strict premonoidal categories. However, it will take some time to reach the definition of the former. So, for concreteness, an example of a strict premonoidal category is given by

Example 3.1. Let M be a monoid. Regard M as a one object category, and consider the map from $M \times M$ to M determined by the multiplication of M. This gives a strict premonoidal category. It is strict monoidal if and only if M is commutative.

For a more substantial example,

Example 3.2. Let C be a small category, and let $[C, C]_u$ denote the category of endofunctors on C and transformations between them. Composition of endofunctors with each other and with transformations endows $[C, C]_u$ with a strict premonoidal structure.

In order to generalise from strict premonoidal categories to premonoidal categories and be sure our concept has some definitive status, we need to generalise the coherence theorem for monoidal categories to premonoidal categories. The coherence theorem allows us to speak unambiguously of maps from an n-fold tensor product. To obtain such a coherence result, we need a little delicacy in defining a premonoidal category. So we say

Definition 3.2. A *binoidal category* is a category C together with a functor $\otimes : C \otimes C \longrightarrow C$.

Spelling this out by unwrapping the definition of $C \otimes C$, a binoidal category amounts to a category C together with, for each object x of C, functors $h_x : C \longrightarrow C$ and $k_x : C \longrightarrow C$ such that for each pair (x, y) of objects of C, $h_x y = k_y x$. The joint value is denoted $x \otimes y$.

Definition 3.3. An arrow $f : x \longrightarrow y$ in a binoidal category is *central* if for every arrow $f' : x' \longrightarrow y'$, the two composites from $x \otimes x'$ to $y \otimes y'$ agree, and the two composites from $x' \otimes x$ to $y' \otimes y$ agree, and will be denoted $f \otimes f'$ and $f' \otimes f$ respectively. (Note that we only use this notation when one of the maps involved is central. It is not well defined for arbitrary maps.)

It follows from the definition that, in the presence of natural associativity of $\otimes$, which will be part of the definition of premonoidal category, if $f : x \longrightarrow y$ is central and z is any object of C, then $h_z(f) : z \otimes x \longrightarrow z \otimes y$ and $k_z(f) : x \otimes z \longrightarrow y \otimes z$ are central.

Definition 3.4. Given a binoidal category C, a natural transformation $\alpha : g \Longrightarrow h : B \longrightarrow C$ is called *central* if every component of α is central.

Observe again that, in the presence of natural associativity of $\otimes$, if $\alpha : g \Longrightarrow h : B \longrightarrow C$ is central and $k : A \longrightarrow C$ is any functor, then $\alpha \otimes k : g \otimes k \Longrightarrow h \otimes k : B \otimes A \longrightarrow C$ is a natural transformation and is central, and dually.

Definition 3.5. A *premonoidal category* is a binoidal category C together with an object I of C, and central natural isomorphisms a with components $(x \otimes y) \otimes z \longrightarrow x \otimes (y \otimes z)$, l with components $x \longrightarrow x \otimes I$, and r with components $x \longrightarrow I \otimes x$, subject to two equations: the pentagon expressing coherence of a, and the triangle expressing coherence of l and r with respect to a.

Observe that every monoidal category is a premonoidal category, as is every strict premonoidal category. Having defined the notion of premonoidal category, we can immediately make a subsidiary definition of fundamental importance, that of the centre of a premonoidal category. This concept generalises that of the centre of a group and yields the easiest proof of coherence for premonoidal categories. It is also needed to characterize strong monads and their various conditions in this setting.

Definition 3.6. Given a premonoidal category C, define the *centre* of C to be the subcategory of C consisting of all the objects of C and the central morphisms.

We denote the centre of a premonoidal category C by $Z(C)$. As mentioned earlier, in the presence of natural associativity of $\otimes$, as we have in the definition of premonoidal category, h_z and k_z preserve central maps. It immediately follows that we have

Proposition 3.1. The centre of a premonoidal category is a monoidal category.

For a simple example of the centre of a premonoidal category,

Example 3.3. If G is a group regarded as a one object premonoidal category as in Example 3.1, the centre of G agrees with the usual definition for a group.

We can now prove coherence for premonoidal categories as follows.

Theorem 3.1. Every diagram built from the structural natural transformations in the definition of a premonoidal category commutes.

Proof. Since the centre of a premonoidal category is a monoidal category and all the structural maps are central, the result follows immediately from coherence for a monoidal category as in Kelly's refinement (Kelly 1964) of Mac Lane's proof. □

We now turn to the definition of a symmetric premonoidal category.

Definition 3.7. A *symmetry* for a premonoidal category is a central natural isomorphism with components $c : x \otimes y \longrightarrow y \otimes x$, satisfying the two conditions $c^2 = 1$ and equality of the evident two maps from $(x \otimes y) \otimes z$ to $z \otimes (x \otimes y)$. A *symmetric* premonoidal category is a premonoidal category together with a symmetry.

Theorem 3.2. Every diagram built from the structural natural transformations in the definition of a symmetric premonoidal category commutes.

Proof. As for premonoidal categories, this is a consequence of the result for symmetric monoidal categories. □

We now define the notion of premonoidal functor. Most of the definition is a routine generalisation of that of monoidal functor. However, we need to justify our demand that premonoidal functors send central maps to central maps. In order to form a composite of premonoidal functors, one needs structural maps to be sent to central maps. In particular, a premonoidal functor is equipped with a structural map of the form $\bar{g} : ga \otimes gb \longrightarrow g(a \otimes b)$, so in order to define a composite, we must demand that all maps that may be of that form be sent to central maps, and the simplest reasonable condition that ensures that that is the case is the condition that centrality be preserved. So we make the following definition.

Definition 3.8. A *premonoidal functor* $(g, \bar{g}, \hat{g}) : C \longrightarrow D$ is a functor $g : C \longrightarrow D$ that sends central maps to central maps, together with central natural transformations with components $\bar{g} : ga \otimes gb \longrightarrow g(a \otimes b)$ and $\hat{g} : I \longrightarrow g(I)$, subject to the three equations expressing coherence with a, l and r. An *oppremonoidal functor* is a premonoidal functor but with the natural transformations in the reverse direction. A premonoidal functor is called *strong* or *strict* when $\bar{g}$ and $\hat{g}$ are isomorphisms or identities respectively.

Theorem 3.3. Every diagram built from the structural natural transformations in the definition of premonoidal functor commutes.

One may similarly generalise the definition and coherence theorem for symmetric monoidal functors to symmetric premonoidal functors and coherence for them. Small premonoidal categories and premonoidal functors form a category. It is routine to define premonoidal natural transformations too, and prove that, together with premonoidal categories and functors, they yield a 2-category. However, we will not need premonoidal natural transformations for our purposes here.

For an example of how these definitions work in practice, consider the modelling of side-effects. The following amounts to a reformulation of Moggi's side-effects monad, with a little extra generality in that the base category need only be assumed to be symmetric monoidal, without any closed structure.

Example 3.4. Let C be a symmetric monoidal category, and let S be an object of C. Define D to be the category with $ObD = ObC$, and $D(x, y) = C(x \otimes S, y \otimes S)$, with the evident composition. Then D has a premonoidal structure such that the canonical functor

$j : C \longrightarrow D$ given by sending an arrow f to $f \otimes S$ is strict premonoidal. Specifically, the tensor operation applied to a pair of objects (x, y) is determined by the monoidal structure of C; and given z and $f : x \otimes S \longrightarrow y \otimes S$, define $z \otimes f$ to be the evident map from $(z \otimes x) \otimes S$ to $(z \otimes y) \otimes S$, and similarly, modulo the symmetry of C, for $f \otimes z$. Observe that the functor $j : C \longrightarrow D$ factors through the centre of D. Moreover, since the structural isomorphisms of D all lie in C, they are central. Observe further that if $C = Set$ and S has at least two elements, then $Z(D) = Set$.

More generally, to see how monads with a strength fit into our setting, consider

Example 3.5. Let C be symmetric monoidal, and let T be a strong monad on C, with K_T the Kleisli category of T. Then K_T has a premonoidal structure making the canonical functor from C to K_T strict premonoidal. We shall prove this, together with a converse, in Section 4. As in Example 3.4, the behaviour of the operation on objects is determined by that on C. For its behaviour on morphisms, given z and $f : x \longrightarrow Ty$, define $z \otimes f$ by the composite $z \otimes x \longrightarrow z \otimes Ty \longrightarrow T(z \otimes y)$, and similarly modulo the symmetry of C for $f \otimes z$. Observe that the functor $j : C \longrightarrow K_T$ factors through $Z(K_T)$. We shall show later that the assertion that the premonoidal structure on K_T is monoidal is equivalent to the assertion that the strong monad is commutative; in other words, T is commutative if and only if $K_T = Z(K_T)$.

Premonoidal structure allows us to model product types, but we also want to model conditionals, and that means we need an account of finite sums. That is easy to add by emulating the definition of distributive category as in (Carboni *et al.* 1993) as follows.

Definition 3.9. A *predistributive* category is a premonoidal category with finite sums, for which $x \otimes -$ and $- \otimes x$ preserve finite sums. A *symmetric predistributive* category is a predistributive category together with a symmetry.

A distributive category is a predistributive category in which the premonoidal structure is monoidal; similarly for symmetric distributive categories. It is evident how to define strong predistributive functors and variants: they are strong premonoidal functors that preserve finite sums and variants. Examples 3.4 and 3.5 provide instances of predistributive categories and strict predistributive functors. The condition that $x \otimes -$ preserves finite sums ensures that the semantics of programs of several variables behaves well in the presence of conditionals, cf (Anderson and Power 1996).

4. Monads and premonoidal categories as notions of computation

To recall from Moggi's paper (Moggi 1991), a monad T is said to satisfy the *mono requirement* if the unit of the monad is a monomorphism pointwise. A tensorial strength for a monad on a monoidal category is a natural transformation with components $t : x \otimes Ty \longrightarrow T(x \otimes y)$ subject to the four coherence conditions expressing coherence with respect to the rest of the data for a monad and the associativity and right unit isomorphisms. We define a *costrength* for a monad on a monoidal category C to be a strength for the monad regarded as a monad on C_t, which is the same category as C but with the tensor twisted. Of course, if C is symmetric monoidal, then to give a costrength for T is equivalent to giving a strength for T. A *bistrength* for a monad on a monoidal category C consists of a strength t and a costrength s, such that the evident two induced maps from $(x \otimes Ty) \otimes z$ to $T((x \otimes y) \otimes z)$ agree. A tensorial strength on a symmetric monoidal category is commutative if the evident two natural transformations with components from $Tx \otimes Ty$ to $T(x \otimes y)$ agree.

The central facts for us in characterizing Moggi's constructions are

Theorem 4.1. (Kleisli) Given a category C, to give a functor $j : C \longrightarrow D$ that is the identity on objects and has a right adjoint is to give a monad on C.

Proof. This is an immediate consequence of the Kleisli construction: given j, the induced monad is that determined by its right adjoint. Conversely, given a monad T, define D to be the Kleisli category for T, with j the inclusion of C into K_T. These constructions are mutually inverse (up to isomorphism of categories). □

Proposition 4.1. Given a monad T on a category C, T satisfies the mono requirement if and only if $j : C \longrightarrow K_T$ is faithful.

Proof. This is an immediate consequence of the definition of j in terms of T. □

The two results above characterize Moggi's computational monads, which are defined to be monads that satisfy the mono requirement, in our terms. Of course, a functor that has a right adjoint necessarily preserves finite sums. So Moggi's basic setup, equivalently a faithful identity on objects functor with right adjoint, is a special case of a structure we would take as primitive, i.e., a faithful functor which preserves finite sums. Our approach in general is to start with a functor, then add structure to the categories and functor: a fundamental structure is one to account for contexts. That leads us to a monoidal category, a premonoidal category, and a strong premonoidal functor from the first to the second. In more special cases, the monoidal structure may be asked to be finite product structure, as for instance in Set or the category of ω-cpo's. The premonoidal structure may be asked to be monoidal, as for instance in modelling concurrency, where a monoidal structure is often used to model parallel composition. We may also ask for the existence and preservation of finite sums as mentioned above.

To illustrate the extra generality we gain by this approach, consider the modelling of concurrency. Recently, Milner and colleagues have developed the concept of a control structure (Mifsud *et al.* 1995) to model concurrency, leading examples being given by variants of the π-calculus and by Petri nets. Modulo two caveats, control structures have been proved equivalent to elementary control structures (Power 1996), which we now define.

Let M denote the free category with strictly associative finite products on a set P. If $P = 1$, then M is equivalent to Set_f^{op}, where Set_f denotes the category of finite sets. To define an elementary control structure, we assume that P is given in advance, as is a set of controls K, each with arity information (the idea being that a control takes any parametrized family of arrows to a parametrized arrow, and the arity information spells out the possible domains and codomains, details appear in (Mifsud *et al.* 1995; Power 1996)).

Definition 4.1. An *elementary control structure* consists of

— a strict symmetric monoidal locally preordered category C,

— an identity on objects strict symmetric monoidal functor $j : M \longrightarrow C_0$ such that each projection $\pi_2 : k \times m \longrightarrow m$ is maximal in $C(k \times m, m)$,
— for each control K with associated arity information $((m_1, n_1), \cdots, (m_r, n_r)) \mapsto (m, n)$ and each k, a function $C_0(k \times m_1, n_1) \times \cdots \times C_0(k \times m_r, n_r) \longrightarrow C_0(k \times m, n)$, natural with respect to maps $f : k \longrightarrow k'$ in M.

It is routine to verify that the naturality condition on controls may be expressed in the form that the given family of functions forms a natural transformation between two functors from M to Set.

This definition of elementary control structure places it immediately as an instance of the structures we consider here. Note that the functor from M into C_0 is not assumed to have an adjoint, and there has been no reason to add that condition in any of the work to date: see (Mifsud *et al.* 1995; Hermida and Power 1995; Power 1996).

Returning to monads, specifically to Moggi's use of the strength of a monad, the strength he introduces corresponds exactly, in the presence of a monad, to symmetric premonoidal structure, as we now make precise. We prove a series of results which allow us to characterize a strength for a monad in terms of a symmetric premonoidal structure, then characterize commutativity for such a strength as the natural condition that the symmetric premonoidal structure be symmetric monoidal.

Proposition 4.2. Let $\eta, \epsilon : j \dashv g : D \longrightarrow C$ and suppose C is a monoidal category, D is premonoidal, and j is strict premonoidal. Let the corresponding monad on C be T. Then T has a bistrength given as follows: $t : x \otimes Ty \longrightarrow T(x \otimes y)$ is the adjoint correspondent of $jx \otimes \epsilon_{jy} : j(x \otimes Ty) = jx \otimes jgjy \longrightarrow jx \otimes jy = j(x \otimes y)$, and the costrength is given dually.

Proof. Routine diagram chasing. □

Theorem 4.2. Let C be a monoidal category, and let T be a monad on it. Then, to give a bistrength for T is to give a premonoidal structure on K_T such that $j : C \longrightarrow K_T$ is a strict premonoidal functor.

Proof. Given a bistrength for T, and given $f : x \longrightarrow Ty$ and z, define $z \otimes f$ by $z \otimes x \longrightarrow z \otimes Ty \longrightarrow T(z \otimes y)$ and define $f \otimes z$ by $x \otimes z \longrightarrow Ty \otimes z \longrightarrow T(y \otimes z)$, using the strength and costrength of T respectively. Two of the axioms of a strength, respectively costrength, plus the compatibility condition, ensure that a and r, respectively a and l, are natural in K_T. That they are central follows from the fact that they are arrows in C, and they satisfy the axioms because they do so in C. The other two axioms for a strength, respectively costrength, ensure that $z \otimes -$, respectively $- \otimes z$, is functorial. For the converse, define $t : x \otimes Ty \longrightarrow T(x \otimes y)$ by $x \otimes id_{Ty}$, where id_{Ty} is regarded as an arrow in K_T from Ty to y. Seen as such, it is the y-component of the counit of the adjunction. Define $s : Tx \otimes y \longrightarrow T(x \otimes y)$ dually. The condition on a strict premonoidal functor that forces j to factor through the centre of K_T forces t and s to be natural.
In proving that these processes are mutually inverse, the tricky part is to verify that, starting with a premonoidal structure on K_T, then applying the processes returns the same premonoidal structure. Denote the monoidal structure on C by $\otimes$ and the premonoidal structure on K_T by $\odot$. To elucidate the proof, we will denote objects of C by x, y and z, and the same objects regarded as objects of K_T by jx, jy and jz. Note that

both the monad and the right adjoint to j are given by T. The two roles of T should be clear from the context, but their sameness is essential. Given x and $f : jy \longrightarrow jz$ in K_T, we must show that $jx \odot f : j(x \otimes y) = jx \odot jy \longrightarrow jx \odot jz = j(x \otimes z)$ corresponds under the adjunction to $x \otimes y \longrightarrow x \otimes Tjz \longrightarrow Tj(x \otimes z)$, where the first map is given by $x \otimes \bar{f}$, where $\bar{f} : y \longrightarrow Tjz$ corresponds under the adjunction to $f : jy \longrightarrow jz$, and the second is the correspondent of $jx \odot (id_{jTjz} : jTjz \longrightarrow jz)$, equivalently $jx \odot \epsilon_{jz}$, given by the counit of the adjunction. Since an adjunction is natural in its domain, this composite is the correspondent of the composite in K_T of $j(x \otimes \bar{f})$ with $jx \odot \epsilon_{jz}$, and that this composite equals $jx \odot f$ follows directly from the facts that j is strict premonoidal, $jx \odot -$ is functorial, and the behaviour of the counit of an adjunction. □

Corollary 4.1. Let C be a symmetric monoidal category with T a monad on it. Then to give a bistrength for T that satisfies the evident compatibility condition with respect to the commutativity isomorphism is to give a monoidal structure on K_T making $j : C \longrightarrow K_T$ strict monoidal.

Proof. By the theorem, it suffices to prove that the premonoidal structure on K_T determined by a bistrength is monoidal if and only if the strength and costrength are compatible with respect to the commutativity. The reverse direction follows by considering $id_{Tx} \otimes id_{Ty}$. The forward direction follows by inspection of the premonoidal structure and by naturality. □

Trivially, to give a costrength on a symmetric monoidal category is equivalent to giving a strength: one can obtain one from the other via two applications of symmetry. Moreover, given a strength, the bistrength condition on it and its corresponding costrength follows from its naturality with respect to the commutativity maps. So we have

Corollary 4.2. Let C be a symmetric monoidal category with T a monad on it. Then, to give a strength for T is to give a symmetric premonoidal structure on K_T that makes j a strict symmetric premonoidal functor.

Proof. The symmetry on K_T ensures that the strength and costrength correspond. Given stength and costrength $t : x \otimes Ty \longrightarrow T(x \otimes y)$ and $s : Ty \otimes x \longrightarrow T(y \otimes x)$, t is $x \otimes id_{Ty}$, and s is $id_{Ty} \otimes x$. □

Corollary 4.2 characterizes Moggi's strong monads in our terms and in fact is a little more general, since Moggi considers only strong monads on a category with finite products. Moreover, by Proposition 4.1, such a strong monad will in addition satisfy Moggi's mono requirement if and only if j is faithful. We can now characterize those monads with commutative strength as follows.

Corollary 4.3. Let C be a symmetric monoidal category with T a monad on it. Then, to give a commutative strength for T is to give K_T a symmetric monoidal structure making $j : C \longrightarrow K_T$ strict symmetric monoidal.

Proof. In the case that a strength and a costrength correspond via the commutativity isomorphism, the compatibility condition of Corollary 4.1 is precisely the commutativity condition for the strength. □

For most of the categories C of primary interest to us, such as *Poset* and the category of ω-cpos with least element, C is symmetric monoidal closed. So to give a strength is to give an enrichment; but typically, as in *Poset* and the category of ω-cpos with least element, there is at most one such. So there is typically only one symmetric premonoidal structure on our category of computations for which the inclusion j is strict premonoidal.

5. Factorization on Cat

In this section, we show that for the study of premonoidal, monoidal, or predistributive structure, no substantial generality is lost in restricting to functors $j : C \longrightarrow D$ that are the identity on objects. We do that by proving that every functor strictly preserving such structure factors as an identity on objects structure preserving functor followed by a fully faithful one.

Theorem 5.1. Let (E, M) be a factorization system on a category C. Let T be a monad on C such that if $e \epsilon E$, then $Te \epsilon E$. Then (E, M) lifts to a factorization system (E', M') on $T - Alg$ for which the forgetful functor to C preserves both $E's$ and $M's$.

Proof. Say $f : (A, a) \longrightarrow (B, b)$ is in E' whenever $f \epsilon E$, and say $f \epsilon M'$ whenever $f \epsilon M$. All isomorphisms are in E' and M', and both are closed under composition. It remains to show that factorizations exist and that the diagonal fill-in property holds.
For the former, given any $f : (A, a) \longrightarrow (B, b)$, let $f = m \cdot e$ be its (E, M)- factorization, with $e : A \longrightarrow C$. Consider $b \cdot Tm \cdot Te = m \cdot e \cdot a$. Since $Te \epsilon E$, the diagonal fill-in for (E, M) provides $c : TC \longrightarrow C$. Since $e \epsilon E$ and $TTe \epsilon E$, applying the unicity property for the diagonal fill-in proves that (C, c) is a T-algebra. It is defined to make e and m maps of algebras.
For the latter, given a commutative diagram $g \cdot e' = m' \cdot f$ in $T - Alg$, by our definitions of E' and M', $e' \epsilon E$ and $m' \epsilon M$. So the diagonal fill-in property supplies unique $h : B \longrightarrow C$ such that $h \cdot e' = f$ and $m' \cdot h = g$. That h is a map of algebras follows from unicity of the diagonal fill-in and the fact that $Te' \epsilon E$. □

Proposition 5.1. The families B of bijective on objects functors and F of fully faithful functors form a factorization system on Cat with the added property that for any diagram of the form $\alpha : g \cdot e \Longrightarrow m \cdot f$ with $(e : C \longrightarrow D) \epsilon B$, with $(m : C' \longrightarrow D') \epsilon F$, and with α an isomorphism, there exist unique $h : D \longrightarrow C'$ and $\beta : m \cdot h \Longrightarrow g$ such that $h \cdot e = f$ and $\beta \cdot e = \alpha$.

Proof. Routine. □

Corollary 5.1. For any monad T on Cat_g, the 2-category of small categories, functors, and natural isomorphisms, such that if $e \epsilon B$, then $Te \epsilon B$, every pseudo-map of algebras $f : (A, a) \longrightarrow (B, b)$ factors uniquely up to unique isomorphism as a strict map that is bijective on objects followed by a pseudo-map that is fully faithful.

Proof. Essentially as for Theorem 5.1. This result is essentially proved in (Power 1989), where it is used to show that every pseudo-T-algebra is equivalent to a strict T-algebra. □

Examples of such monads are those whose algebras are monoidal categories, symmetric monoidal categories, categories with finite sums, and (symmetric) distributive categories.

We also require the well known and easily verifiable result

Proposition 5.2. Given any functor $f : C \longrightarrow D$ with right adjoint g, let $f = h \cdot j$ be the (B, F)-factorization of f. Then j has a right adjoint $g \cdot h$. Thus, f factors through the Kleisli category of $g \cdot f$.

For definiteness, we choose a (B, F)-factorization of any functor f as $f = h \cdot j$ with j the identity on objects.

These results justify the particular attention we pay to functors $j : C \longrightarrow D$ that are the identity on objects, rather than any functor, respecting sums and monoidal structure. Although not an instance of Corollary 5.1, it is also true that the ordinary monad for premonoidal categories preserves $B's$, so allows a strict premonoidal functor to be factored as one that is the identity on objects followed by a fully faithful one. The same is true for all our variants such as symmetric predistributive categories.

Proposition 5.3. Let $f = h \cdot j$ be a (B, F)-factorization. Then f is faithful if and only if j is faithful.

Proof. Trivial. □

Finally, observe that all of the above results enrich without difficulty: so this account extends to enrichment over the category of ω-cpo's with least element, which enables us to model recursion.

6. A representation theorem

Here, we should like to show that any small faithful identity on objects strict predistributive functor embeds into the left adjoint part of the Kleisli construction of a strong monad that satisfies the mono requirement on a category, preferably a small category. In fact, we cannot yet prove such a result in the full generality of symmetric premonoidal structure, so restrict ourselves to symmetric monoidal structure: but we do correspondingly strengthen the conclusion to find a commutative monad. Moreover, we can only prove the faithfulness part under a mild extra condition. Subject to those reservations, we can prove the result. This shows that formally we do not gain substantial generality in passing from monads as notions of computation to our setting for modelling notions of computation, although we will give some examples to show we gain some insight. The value of our premonoidal category analysis lies not so much in its added generality as in the new choice of mathematical primitives.

Theorem 6.1. Let C and D be small categories and let $j : C \longrightarrow D$ be the identity on objects. Then there exist fully faithful functors $in_C : C \longrightarrow \bar{C}$ and $in_D : D \longrightarrow \bar{D}$ together with an identity on objects functor $\bar{j} : \bar{C} \longrightarrow \bar{D}$ making the induced square commute, and with $\bar{j}$ having a right adjoint. Moreover, we can choose $\bar{C}$ and $\bar{D}$ small.

Proof. Let $\bar{C}$ be the free cocompletion of C, i.e., the functor category $C^{op} \longrightarrow Set$. Then, since $D^{op} \longrightarrow Set$ is cocomplete, j extends to a functor $\hat{j} : \bar{C} \longrightarrow (D^{op} \longrightarrow Set)$ with a right adjoint, the latter given by $j^{op} \longrightarrow Set$. We have enough choice in describing $\hat{j}$ to ensure that it commutes strictly with j. Of course, $\hat{j}$ may not be the identity on objects. So define $\bar{j}$ and $\bar{D}$ by the (B, F)-factorization of $\hat{j}$. By our choice of factorization, $\bar{j}$ is the identity on objects. We have in_D by the diagonal fill-in property. By Proposition 5.2, $\bar{j}$

has a right adjoint. Finally, we can force $\bar{C}$ and $\bar{D}$ to be small by taking the smallest full subcategories of $\bar{C}$ and $\bar{D}$ containing C and D respectively and closed under the monad or comonad respectively. $\square$

Corollary 6.1. Let C and D be small (symmetric) monoidal categories, with identity on objects strict (symmetric) monoidal $j : C \longrightarrow D$. Then there exist small (symmetric) monoidal categories $\bar{C}$ and $\bar{D}$, fully faithful strict (symmetric) monoidal functors $in_C : C \longrightarrow \bar{C}$ and $in_D : D \longrightarrow \bar{D}$ together with an identity on objects strict (symmetric) monoidal functor $\bar{j} : \bar{C} \longrightarrow \bar{D}$ making the induced square commute, and with $\bar{j}$ having a right adjoint.

Proof. This follows from the theorem by use of the fact that $C^{op} \longrightarrow Set$ is the free (symmetric) monoidal cocompletion of a small (symmetric) monoidal category C (see (Im and Kelly 1986)). $\square$

Observe that in the corollary, the monoidal structure on $\bar{C}$ induced by that on C makes $\bar{C}$ monoidal biclosed. The corollary shows that j embeds into the left adjoint part of the Kleisli construction for a commutative monad.

Theorem 6.2. Given small symmetric distributive categories C and D and identity on objects strict symmetric distributive $j : C \longrightarrow D$, j embeds into the left adjoint part of the Kleisli construction for a commutative monad on a small symmetric monoidal closed category.

Proof. This is essentially the same proof as that for Theorem 6.1 and Corollary 6.1. The only modifications we need to make to the proof of the former are to replace the free cocompletion by the free cocompletion that respects finite sums, i.e., replace $C^{op} \longrightarrow Set$ by $FP(C^{op} \longrightarrow Set)$, the category of finite product preserving functors from C^{op} to Set (see (Kelly 1982) Theorems 5.86 and 6.11), and similarly for D, and in forcing $\bar{C}$ and $\bar{D}$ to be small, close under the monoidal closed structure and finite sums. The distributivity condition is necessary because $x \otimes -$ preserves colimits, and hence has a right adjoint, if and only if its restriction to C preserves finite sums, equivalently distributivity. $\square$

Theorem 6.3. Let C and D be small symmetric distributive categories, with $j : C \longrightarrow D$ the identity on objects, strict symmetric distributive and faithful. Assume moreover that C has and $x \otimes -$ and j preserve finite colimits. Then, j embeds into the Kleisli construction for a commutative monad that satisfies the mono requirement on a small symmetric monoidal closed category.

Proof. The construction is as for Theorem 6.1 except for replacing the free cocompletion of C by the free cocompletion that respects finite colimits, and replacing the free cocompletion of D by the free cocompletion that respects the finite colimits of those diagrams that factor through j. Then, the proofs of Theorem 6.1 and Corollary 6.1 generalise easily. By Proposition 5.3, it remains to show that $\bar{j}$ is faithful. To do that, one uses the fact that the free cocompletion that respects finite colimits is $FL(C^{op} \longrightarrow Set)$, which is locally finitely presentable. So filtered colimits commute with finite limits. Each object of $\bar{C}$ is a filtered colimit of a diagram in C. Moreover, each object x of C is finitely presentable in $\bar{C}$, so $\hat{j}x$ is finitely presentable in $D^{op} \longrightarrow Set$. Now suppose $\hat{j}h = \hat{j}k : \hat{j}a \longrightarrow \hat{j}b$. Each $h\alpha_x$ with $a = colim(\alpha)$ factors through some β_y, where

$b = colim(\beta)$. By filteredness, we may assume $\hat{j}h\alpha_x$ and $\hat{j}k\alpha_y$ factor through the same $j\beta_y$, and by filteredness again, both are equal as maps into $j\beta_y$. So, since j is faithful, we are done. □

We have shown that, modulo some side conditions, our general setting of a pair of categories with structure and a structure preserving functor $j : C \longrightarrow D$ that is the identity on objects embeds into the Kleisli construction for a monad with structure. So in a sense, we gain little extra generality from our setting. However, that is somewhat illusory. For instance, in Example 4.1, the functor from M to C_0 did not in general have a right adjoint, and formally extending an elementary control structure in order to force an adjoint to exist does not seem obviously warranted by analysis of concurrency. Another example in which embedding to force a right adjoint does not seem obviously warranted is given by continuations as follows.

Example 6.1. As a first attempt to model continuations, one may have a small category C as base category, then for category D of continuations, have objects the same as those of the base category, with a morphism from x to y being an ObC-indexed family of morphisms from $y \longrightarrow z$ to $x \longrightarrow z$. For size reasons, the inclusion j will not have a right adjoint in general. By Theorem 6.1, we can embed j into a functor that does have a right adjoint, but that adjoint has a different spirit to that of the relationship between C and D as it involves a product, not over $Ob\bar{C}$ but over ObC.

Continuations will be analysed by means of premonoidal categories in the thesis of Hayo Thielecke. He gives a considerably more delicate account of continuations than in this example, but a premonoidal category is still the basic structure he requires.

7. A fibrational view

In this section, we show that premonoidal categories allow the construction of what has been called the simple fibration: this construction has usually been applied only to categories with finite products, as superficially it appears that one requires finite products in order to make the construction. The fact that we can make this construction and reformulate our leading result in these terms offers support to the idea that premonoidal categories are an appropriate structure with which to model contexts: they seem a natural general setting for the fibrational construction and analysis.

Definition 7.1. A *monoid* in a premonoidal category C consists of an object m of C, and central maps $\mu : m \otimes m \longrightarrow m$ and $\iota : I \longrightarrow m$, making the usual associativity and unit diagrams commute.

It follows from centrality of the two maps in the definition of monoid that one has the usual coherence for a monoid, i.e., n-fold associativity is well defined, and multiple products with units are also well defined.

Definition 7.2. A *monoid map* from m to n in a premonoidal category C is a central map $f : m \longrightarrow n$ that commutes with the multiplication and units of the monoids.

Again, it follows from centrality that a monoid map preserves multiple application of multiplication and units. Given a premonoidal category C, monoids and monoid maps in C form a category $Mon(C)$ with composition given by that of C.

Trivially, any monoid m in a premonoidal category C yields a monad on C given by $m \otimes -$, and any monoid map $f : m \longrightarrow n$ yields a map of monads from $m \otimes -$ to $n \otimes -$, and hence a functor from $K_{m\otimes -}$, the Kleisli category of the monad $m \otimes -$, to $K_{n\otimes -}$, that is the identity on objects.

Dualizing, given a premonoidal category C, a *comonoid* in C is defined to be a monoid in C^{op}, and the above construction dualizes, so that any comonoid c yields a comonad $c \otimes -$, and a Kleisli category $K_{c\otimes -}$ for the comonad, all this functorially. So letting $Comon(C)$ denote the category of comonoids in C, we have a functor from $Comon(C)^{op}$ to Cat, which we denote by s_C.

If the premonoidal structure of C is given by finite products, each object c of C has a unique comonoid structure, given by the diagonal and the unique map to the terminal object, so $Comon(C)$ is isomorphic to C, and s_C amounts to a fibration over C. This fibration has sometimes been called the simple fibration over C, e.g., (Hermida and Jacobs 1995).

So, given a category C with finite products, a premonoidal category D, and a strict premonoidal functor $j : C \longrightarrow D$, the functor j factors through $Comon(D)$: with mild abuse of notation, we denote the induced functor from C to $Comon(D)$ by j too. There is a canonical natural transformation $\tilde{j} : s_C \Longrightarrow s_D \cdot j : C^{op} \longrightarrow Cat$, equivalently a canonical strict map of fibrations from s_C to the pullback fibration of s_D along j: on the fibre over 1, it is j. Thus we have

Proposition 7.1. An identity on objects strict premonoidal functor $j : C \longrightarrow D$ has a right adjoint if and only if $\tilde{j} : s_C \Longrightarrow s_D \cdot j : C^{op} \longrightarrow Cat$ has a right adjoint.

Proof. Since j is the identity on objects, any arrow of $(s_D \cdot j)(x)$ is of the form $f : j(x \times y) \longrightarrow z$. The forward direction of the proposition follows immediately by naturality. The converse follows by consideration of the natural transformation $\tilde{j}$ evaluated at 1. □

This corresponds to the result outlined in (Moggi 1991) to the effect that to give a strong monad on a category C with finite products is to give a monad on the fibration s_C. Putting that together with the above yields

Theorem 7.1. Given a category C with finite products, the following are equivalent:

1 a symmetric premonoidal category D together with an identity on objects strict symmetric premonoidal functor $j : C \longrightarrow D$ that has a right adjoint
2 a symmetric premonoidal category D together with an identity on objects strict symmetric premonoidal functor $j : C \longrightarrow D$ such that $\tilde{j} : s_C \Longrightarrow s_D \cdot j : C^{op} \longrightarrow Cat$ has a right adjoint
3 a monad on s_C
4 a strong monad on C.

Proof. Proposition 7.1 shows the equivalence of 1 and 2. The equivalence of 3 and 4 is routine, given the definition of strength and the construction of s_C. The equivalence of 1 and 4 amounts to Corollary 4.2. □

References

Anderson, S. O. and Power, A. J. (1996) A representable approach to finite nondeterminism. In *Proc. MFPS '94, Theoretical Computer Science* (to appear).

Carboni, A., Lack, S. and Walters, R. F. C. (1993) Introduction to extensive and distributive categories. *J. Pure Appl. Algebra* **84**, 145–158.

Fiore, M. (1996) Axiomatic domain theory in categories of partial maps. *Cambridge University Press Distinguished Dissertations in Computer Science.*

Fiore, M. and Plotkin, G. D. (1994) An axiomatization of computationally adequate domain theoretic models of *FPC*. In *Proc. 9th LICS conference*, 92–102.

Hermida, C. and Jacobs, B. (1995) Fibrations with indeterminates: contextual and functional completeness for polymorphic lambda calculi. *Math. Struct. in Comp. Science* **5**, 501–531.

Hermida, C. and Power, A. J. (1995) Fibrational control structures. In *Proc. CONCUR '95: Concurrency theory, Lecture Notes in Computer Science* **962**, 117–129.

Im, G. B. and Kelly, G. M. (1986) A universal property of the convolution monoidal structure. *J. Pure Appl. Algebra* **43**, 75–88.

Kelly, G. M. (1964) On Mac Lane's conditions for coherence of natural associativities, commutativities, etc. *J. Algebra* **1**, 397–402.

Kelly, G. M. (1982) *Basic concepts of enriched category theory. London Math. Soc. Lecture Notes Series* **64**. Cambridge University Press.

Moggi, E. (1991) Notions of computation and monads. *Information and Computation* **93**, 55–92.

Mifsud, A., Milner, R. and Power, A. J. (1995) Control structures. In *Proc. 10th LICS conference*, 188–198.

Power, A. J. (1989) A general coherence result. *J. Pure Appl. Algebra* **57**, 165–173.

Power, A. J. (1996) Elementary control structures. In *Proc. CONCUR '96: Concurrency theory, Lecture Notes in Computer Science*, 115–130.

Power, A. J. and Robinson, E. P. (1995) On the categorical semantics of types. Draft.