ELSEVIER

Theoretical Computer Science 278 (2002) 303–321

Theoretical Computer Science

www.elsevier.com/locate/tcs

Premonoidal categories as categories with algebraic structure

John Power [1]

Division of Informatics, University of Edinburgh, King's Buildings, Edinburgh EH9 3JZ, UK

Dedicated to Peter Freyd on the occasion of his 60th birthday

Abstract

We develop the study of premonoidal categories. Specifically, we reconcile premonoidal categories with the usual study of categories with algebraic structure by adding a little extra structure. We further give a notion of closedness for a premonoidal category with such extra structure, and show that every premonoidal category fully embeds into a closed one.

1. Introduction

The notion of premonoidal category was introduced in [1, 16] in order to give a unified account of what have been called notions of computation, as introduced by Eugenio Moggi in [10]. The idea of a premonoidal category is that it is a mild generalisation of the concept of monoidal category: 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 in the precise sense of Mac Lane's book [9], i.e., given maps $f: x \to y$ and $f': x' \to y'$, the evident two maps from $x \otimes x'$ to $y \otimes y'$ may differ.

A simple example of a premonoidal category arises in analysing side-effects: one may model a programme from A to B by a function from $S \otimes [A]$ to $S \otimes [B]$, where S is the set of states and $[A]$ and $[B]$ denote types of values. Given another programme, to be modelled by a function from $S \otimes [A']$ to $S \otimes [B']$, one obtains two different functions from $S \otimes [A] \otimes [A']$ to $S \otimes [B] \otimes [B']$, either of which could model the composite of the programmes, dependent upon the order in which they are performed. One cannot describe such behaviour in a monoidal category of denotations of types and programmes owing to the presence of bifunctoriality in the monoidal operation. A similar situation arises in modelling nondeterminism by trees [1]. More generally, for any

[1] This work is supported by EPSRC grant GR/J84205: Frameworks for programming language semantics and logic.

0304-3975/02/$ - see front matter
PII: S0304-3975(00)00340-6

strong monad T on a symmetric monoidal category, the Kleisli category for the monad acquires a premonoidal structure (see [16]), and indeed strong monads can be characterized in such terms. So one drops the bifunctoriality requirement in the definition of monoidal category to obtain a premonoidal category: we recall details of the definition in Section 4.

Not all premonoidal categories arise directly from monads. For instance, the notion of $\otimes\neg$-category [17, 19, 20] has been developed over recent years to model higher order structure with control, in particular continuations. A $\otimes\neg$-category is defined to be a symmetric premonoidal category C to model contexts, together with a functor $\neg : C^{op} \to C$ to model a continuation-type constructor, subject to axioms (see Section 4). For instance, it models *ML*-style continuations; and it has been used to refute long-standing conjectures about continuations [19]. Premonoidal categories have also been used to provide a semantic universe of call-by-value computation and to prove a full abstraction result in terms of game semantics [5]. There, the premonoidal category *CBV* is defined to be the category of cbv-types and innocent strategies, not as the Kleisli category for a monad (see Section 4). In general, if one starts with a simple type theory and a call-by-value operational semantics, then takes the fully abstract model, one obtains a premonoidal category [15].

In this paper, we further develop the concept of premonoidal category. There were two problems left outstanding in the original papers. First, there was no analysis of the concept of closedness for a premonoidal category. One has well understood notions of cartesian closed category and closed monoidal category; so similarly, in order to model higher-order types, one seeks a definition and analysis of the concept of closed premonoidal category. Second, in defining the notion of premonoidal category, the hope was that it would lie alongside other basic category theoretic structures, such as those of monoidal category, cartesian category, or distributive category, and could be used much as those concepts are. However, there was no substantial analysis that unifies the concept of premonoidal category with those other concepts. An answer to the second question helps us to answer the first, so we will address them in reverse order.

In giving a unified account of categories with structure, a fundamental fact linking the three examples cited above is that each of them may be seen as a 2-category of algebras for a finitary 2-monad on the 2-category *Cat*. For instance, there is a finitary 2-monad on *Cat* for which the 2-category of algebras is the 2-category of small monoidal categories and strict monoidal functors. There has been a string of papers developing categories with such algebraic structure; an account directed towards computer scientists is in [12]. However, as we explain in Section 4, there seems to be no natural functor from the category of small premonoidal categories and strict premonoidal functors to *Cat* that presents the former as the 2-category of algebras for a finitary 2-monad on *Cat*. However, we can take a mild and natural generalisation of the 2-category *Cat*, and demonstrate that a mild extension of the category of small premonoidal categories is finitarily monadic over it: that is the substance of Section 5. All finitary 2-monads on *Cat* extend too, so this considerably strengthens the relationship between premonoidal categories and other apparently similar category theoretic structures. This requires considerable category theory: some basic work on enriched categories in Section 2, together with the study of algebraic structure on 2-categories in Section 3.

As mentioned above, the algebraic structure of premonoidal categories helps us to define the notion of closed premonoidal category, which we investigate in Section 6. It seems has been used [6, 20, 19] for the study of higher-order types with a call by value evaluation strategy. We give some examples and prove an embedding result.

2. Enriched categories

For a cartesian closed locally small category V, one may speak of *V-categories*, *V-functors* and *V-natural transformations*, yielding the 2-category V-*Cat* of small V-categories. The idea of a V-category C is that it has a set of objects, just as an ordinary category does; but rather than having hom-sets, for each pair of objects x and y of C, it has a hom-object $C(x, y)$ of V; the cartesian structure of V suffices to define composition of a V-category. A V-category is sometimes called a category enriched over V, and the canonical reference, albeit in somewhat greater generality than that here, is Kelly's book [7]. For more examples and analysis directed towards a computing readership, see Robinson's paper [18]. One may see enriched categories in use in Fiore's book [3].

The archetypal example is given by $V = Set$, in which case a V-category is precisely a locally small category, and a small V-category is precisely a small category. Another important example, which we use heavily here, is that of $V = Cat$. Then, a small V-category is precisely a small 2-category.

Every V-category C has an underlying ordinary category C_0: an object of C_0 is an object of C, and an arrow from x to y in C_0 is an arrow in V from the unit of V to $C(x, y)$. In general, not every ordinary category may be seen as the underlying ordinary category of a V-category, but sometimes, there may be many ways of enriching an ordinary category with the structure given by V. In particular, the category V always enriches to a V-category with hom-objects given by the exponentials $[x, y]$ in V [7].

If V is complete, as it will be in all examples of interest to us, then V-*Cat* is complete as a 2-category, and if V is also cocomplete, then V-*Cat* is cocomplete as a 2-category too.

In this paper, we use ordinary categories, i.e., categories enriched over *Set*, and 2-categories, i.e., categories enriched over *Cat*; but our leading two examples of cartesian closed categories over which we wish to enrich are as follows.

Example 1. Consider $V = [\to, Set]$, the functor category, an object of which is a function, and an arrow a commutative square in the category *Set*. A small V-category amounts to a pair of small categories C and D together with an identity on objects functor $j : C \to D$. We will call C the *domain* category of the pair, and D the *codomain* category. A V-functor amounts to a commutative square of functors. A V-natural transformation is a natural transformation between the codomain categories but with each component in the domain category. □

The category $[\rightarrow, Set]$ is complete and cocomplete, with limits and colimits given pointwise, so $[\rightarrow, Set]$-*Cat* is complete and cocomplete. In fact, $[\rightarrow, Set]$-*Cat* is a locally finitely presentable 2-category. Observe that there is a diagonal functor $\Delta : Cat \rightarrow [\rightarrow, Set]\text{-}Cat$, and it is fully faithful with left and right adjoint, thus exhibiting *Cat* as a full reflective and coreflective sub-2-category of $[\rightarrow, Set]$-*Cat*. It follows that any finitary 2-monad on *Cat* extends to a finitary 2-monad on $[\rightarrow, Set]$-*Cat* with the same 2-category of algebras. Note that the underlying ordinary category of a $[\rightarrow, Set]$-category is its domain category.

Example 2. Consider $V = Subset$, an object of which is a set together with a subset, and an arrow a function that respects the subset structure. A small V-category amounts to a small category D together with a subcategory C with the same objects. We will denote the inclusion by $j : C \rightarrow D$, and again call C the *domain* category and D the *codomain* category. A V-functor amounts to a functor that respects the subcategory structure, and a V-natural transformation is a natural transformation with each component in the subcategory. □

The category *Subset* is complete and cocomplete, with limits given pointwise and colimits a little more complex: first take colimits pointwise, then factor the resulting function into a quotient followed by a subset. Again, it follows that *Subset*-*Cat* is complete and cocomplete. It is also a locally finitely presentable 2-category; there is a diagonal functor $\Delta : Cat \rightarrow Subset\text{-}Cat$ with left and right adjoint, exhibiting *Cat* as a full reflective and coreflective subcategory of *Subset*-*Cat*. Thus, every finitary 2-monad on *Cat* extends to a finitary 2-monad on *Subset*-*Cat* with the same 2-category of algebras. (See Section 3 for the significance of finitary monads.) The underlying ordinary category of a *Subset*-category is the domain category.

Although not explicitly studied before, these structures have appeared in computer science, for instance in giving models, called elementary control structures, of Milner's action calculi [11, 13]. To recall the definition of elementary control structure, let M denote the free category with strictly associative finite products on a set P. Assume we have a set of controls K, each with arity information, the idea being that a control takes any parametrised family of arrows to a parametrised arrow, and the arity information spells out the possible domains and codomains: details appear in [13].

Definition 3. An *elementary control structure* consists of a strict symmetric monoidal locally preordered category D and an identity on objects strict symmetric monoidal functor $j : M \rightarrow D_0$, where D_0 is the underlying ordinary category of D, such that each projection $\pi_2 : k \times m \rightarrow m$ is maximal in $D(k \times m, m)$, together with, for each control K with associated arity information $((m_1, n_1), \ldots, (m_r, n_r)) \mapsto (m, n)$ and each k, a function

$D_0(k \times m_1, n_1) \times \cdots \times D_0(k \times m_r, n_r) \to D_0(k \times m, n)$, natural with respect to maps $f : k \to k'$ in M.

If we forget the preorder, then this definition is a special case of the structure we consider here and in later sections, together with controls. 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*. Note that the functor from M into D_0 is not assumed to have an adjoint. So we have

Example 4. Given an elementary control structure, the functor $j : M \to D_0$ is a [→, *Set*]-category. It need not be a *Subset*-category, and it need not have a right adjoint.

For another example, Honda and Yoshida [5] have given a fully abstract model for call-by-value *PCF* in terms of games as follows. They define a notion of cbv-type to be a special kind of sorting. They then define the notion of an innocent strategy, $\sigma : A \to B$, where A and B are cbv-types. They define a composition for innocent strategies, prove that it satisfies the axioms for a category, and call an innocent strategy σ total if $\sigma; \tau = \perp$ implies $\tau = \perp$. The total innocent strategies form a subcategory of the category of innocent strategies. Thus we have

Example 5. Let CBV denote the category of cbv-types and innocent strategies, and let CBV_t denote the category of total innocent strategies. Let $j : CBV_t \to CBV$ be the inclusion. It is therefore a *Subset*-category, hence also a [→, *Set*]-category. Moreover, it has a right adjoint.

Finally, Thielecke [20] defined a $\otimes\neg$-category to be a symmetric premonoidal category (which we shall define in Section 4) C together with a functor $\neg : C^{op} \to C$ satisfying coherence conditions with respect to the premonoidal structure, and such that $\neg$ is self-adjoint on the left. As we shall see, for any premonoidal category C, one has a subcategory $Z(C)$ containing all the objects of C, called the centre of C. The centre was fundamental to Thielecke's analysis of effect-free terms in [19]. We have

Example 6. For any $\otimes\neg$-category C, the inclusion of $Z(C)$ into C is a *Subset*-category, so also a [→, *Set*]-category. It follows from the axioms that j has a right adjoint.

3. Algebraic structure

In this section, we will define *algebraic structure* on the 2-categories *Cat* and our leading examples, Examples 1 and 2. One can define algebraic structure on any locally finitely presentable 2-category, but we would need to introduce the concept of tensor in a 2-category to do so. In the cases of primary interest to us, Examples 1 and 2 and *Cat*, we can avoid introducing tensors. So we avoid them. In what follows, we let C

denote any of these three 2-categories, with Δ defined as in our two examples, and if $C = Cat$, take Δ to be the identity 2-functor. Finally, we refer to finitely presentable objects. In order to understand this paper, one does not need a precise understanding of finitely presentable objects: all one needs to know is that in each of our three examples, they include all finite objects of the 2-category being considered. The only reason we mention finitely presentable objects here is in order to give an accurate statement of the converse of the theorem; but we do not use the converse. For a gentler account of algebraic structure, see Robinson's paper [18].

Let $ob\, C_{\mathrm{f}}$ denote the discrete 2-category on the set of (isomorphism classes of) finitely presentable objects in C. Then a *signature* on C consist of a 2-functor $S : ob\, C_{\mathrm{f}} \to C$. For each $c \in ob\, C_{\mathrm{f}}$, $S(c)$ is called the object of *basic operations of arity* c.

From S, we construct $F(S) : C_{\mathrm{f}} \to C$ as follows: set

$$S_0 = J, \quad \text{the inclusion of } C_{\mathrm{f}} \text{ in } C,$$

$$S_{n+1} = J + \sum_{d \in obC_{\mathrm{f}}} \Delta C(d, S_n(-)) \times S(d),$$

and define

$$\sigma_0 : S_0 \to S_1 \quad \text{to be} \quad inj : J \to J + \sum_{d \in obC_{\mathrm{f}}} \Delta C(d, S_0(-)) \times S(d),$$

$$\sigma_n = J + \sum_{d \in obC_{\mathrm{f}}} \Delta C(d, \sigma_{n-1}(-)) \times S(d) : S_n \to S_{n+1},$$

$$F(S) = colim_{n<\omega} S_n.$$

In many cases of interest, each σ_n is a monomorphism, so $F(S)$ is the union of $\{S_n\}_{n<\omega}$. For each c, we call $F(S)(c)$ the object of *derived* c*-ary operations*.

A signature is typically accompanied by equations between derived operations. So we define the *equations* of an algebraic theory with signature S to consist of a 2-functor $E : ob\, C_{\mathrm{f}} \to C$ together with 2-natural transformations $\tau_1, \tau_2 : E \to F(S)(K(-))$, where $K : ob\, C_{\mathrm{f}} \to C_{\mathrm{f}}$ is the inclusion.

Algebraic structure on C consists of a signature S, together with equations (E, τ_1, τ_2). We will generally denote algebraic structure by (S, E), suppressing τ_1 and τ_2.

We now define the algebras for a given algebraic structure. Given a signature S, an S*-algebra* consists of an object A of C together with a map $v_c : C(c, A) \to C(S(c), A)$ for each c. So, an S-algebra consists of a carrier A and an interpretation of the basic operations of the signature. This interpretation extends canonically to the derived operations, giving an $F(S)(K(-))$-algebra, as follows:

$$v_0 : C(c, A) \to C(S_0(c), A) \quad \text{is the identity},$$

to give $v_{n+1} : C(c, A) \to C(S_{n+1}(c), A)$, using the fact that $C(-, A)$ preserves colimits, is to give a map $C(c, A) \to C(c, A)$, which we will make the identity, and for each d in

$ob\, C_{\rm f}$, a map

$$C(c,A) \to [C(d,S_n(c)), C(S(d),A)],$$

or equivalently, $C(c,A) \times C(d,S_n(c)) \to C(S(d),A)$, which is given inductively by

$$C(c,A) \times C(d,S_n(c)) \stackrel{v_n \times id}{\to} C(S_n(c),A) \times C(d,S_n(c)) \stackrel{\rm comp}{\to} C(d,A) \stackrel{v}{\to} C(S(d),A).$$

Given signature S and equations E, an (S,E)-*algebra* is an S-algebra that satisfies the equations, i.e., an S-algebra (A,v) such that both legs of

$$C(c,A) \stackrel{v_\omega}{\to} C(F(S)(Kc),A) \underset{C(\tau_{2_c},A)}{\overset{C(\tau_{1_c},A)}{\rightrightarrows}} C(E(c),A)$$

agree.

Given (S,E)-algebras (A,v) and (B,δ), we define the hom-category (S,E)-$Alg((A,v),(B,\delta))$ to be the equaliser in Cat of

$$\begin{array}{ccc} C(A,B) & \xrightarrow{\{C(c,-)\}_{c\in obC_{\rm f}}} & \prod_c [C(c,A), C(c,B)] \\ {\scriptstyle \{C(S(c),-)\}_{c\in obC_{\rm f}}} \downarrow & & \downarrow {\scriptstyle \prod_c [C(c,A),\delta_c]} \\ \prod_c [C(S(c),A), C(S(c),B)] & \xrightarrow[\prod_c [v_c, C(S(c),B)]]{} & \prod_c [C(c,A), C(S(c),B)]. \end{array} \qquad (1)$$

This agrees with our usual universal algebraic understanding of the notion of homomorphism of algebras, internalising it to Cat. (S,E)-Alg can then be made into a 2-category in which composition is induced by that in C. An arrow in (S,E)-Alg is an arrow $f: A \to B$ in C such that for all finitely presentable c, $f v_c(-) = \delta_c(f-) : C(c,A) \to C(S(c),B)$, i.e., an arrow in C that commutes with all basic c-ary operations for all c.

The special case of the main result of [8] that is central to our work is

Theorem 7. *A* 2-*category is equivalent to* (S,E)-*Alg for algebraic structure* (S,E) *on* C *if and only if there is a finitary* 2-*monad* T *on* C *such that the* 2-*category is equivalent to* T-*Alg.*

Example 8. We shall see how the category of small categories with binary products is given by algebraic structure on Cat. So let $C = Cat$. Let 2 denote the discrete category on two objects; let $\to$ denote the arrow category; let *Cone* denote the category given by two objects together with a cone over them; and let *Doublecone* denote *Cone* together with a cone over it, i.e., a pair of objects, two cones over the pair of objects, and an intermediary map from the vertex of one of the cones to the vertex of the other, commuting with the projections. Now define $S : ob\, Cat_{\mathrm{f}} \to Cat$ by $S(2) = Cone$, $S(Cone) = Doublecone$, and for all other $c, S(c)$ is the empty category.

An S-algebra is a small category A together with a functor $\phi : [2, A] \to [Cone, A]$ and a functor $\psi : [Cone, A] \to [Doublecone, A]$. The functor ϕ is to take a pair of objects to its limiting cone, and the functor ψ is to take a cone to itself, the limiting cone, and the unique comparison map. So we add equations as follows: we may add equations factoring through $S_1(2)$ and $S_1(Cone)$, respectively, so that $\phi(x) : Cone \to A$ restricts along the inclusion $2 \to Cone$ to x, and so that ψ sends a cone $\sigma : Y \to x$ to a commutative diagram of the form

$$
\begin{array}{ccccc}
Y & & \xrightarrow{\sigma} & & x \\
& {\scriptstyle \gamma_\sigma}\searrow & & \nearrow{\scriptstyle \phi(x)} & \\
& & X & &
\end{array}
$$

Finally, we add an equation factoring through $S_2(2)$ so that, for each $x : 2 \to A$, we have $\gamma_{\phi(x)} = id_X$.

Putting this together, we put $E(2) = Cone + \to$, $E(Cone) = Cone + Cone$, and $E(c)$ to be the empty category for all other c, and we define τ_1 and τ_2 to force the equations as described above: on most components, the τ's factor through $S_1(c)$, but for one of them, we need to factor through $S_2(c)$.

It then follows that for any $x : 2 \to A$, $\phi(x)$ is a limiting cone: given any cone $\sigma : Y \to x$, the diagram $\psi(\sigma)$ provides a comparison map; and given any comparison map $f : Y \to X$, functoriality of ψ applied to the arrow

$$
\begin{array}{ccc}
Y & \xrightarrow{\sigma} & x \\
{\scriptstyle f}\downarrow & & \downarrow{\scriptstyle id_x} \\
X & \xrightarrow[\phi(x)]{} & x
\end{array}
$$

in $[Cone, A]$ shows that

$$
\begin{array}{ccc}
Y & \xrightarrow{\gamma_\sigma} & X \\
{\scriptstyle f}\downarrow & & \downarrow{\scriptstyle id_X} \\
X & \xrightarrow[\gamma_{\phi(x)} = id_X]{} & X
\end{array}
$$

commutes, so $f = \gamma_\sigma$.

So an (S, E)-algebra is precisely a category with assigned binary products. Observe that an (S, E)-algebra map is a functor that sends assigned binary products to assigned binary products.

It is routine to extend the above example to describe small categories with finite products, and a dual gives small categories with finite coproducts; one can also describe small monoidal categories and symmetric monoidal categories. We should like to show that small premonoidal categories can be described in these terms, but as we explain in the next section, they cannot: thus our need to extend from *Cat* to $[\rightarrow, Set]$-*Cat* or *Subset*-*Cat*.

We shall see that we need to pass from *Cat* to a variant in order to incorporate premonoidal categories into the study of categories with algebraic structure. But in addition to that, passing from *Cat* to $[\rightarrow, Set]$-*Cat* is helpful in analysing the structure of elementary control structures as in Example 4. Recall that in an elementary control structure $j : M \rightarrow D_0$, the category M has strictly associative finite products, D_0 is strict symmetric monoidal, and j is a strict symmetric monoidal functor, and that constitutes the type structure of the definition. We can, therefore, define the type structure in the definition of elementary control structure by observing that we have

Example 9. The 2-category of small $[\rightarrow, Set]$-categories $j : C \rightarrow D$ for which C has strictly associative finite products, D is strict symmetric monoidal, and j is strict symmetric monoidal, is given by algebraic structure on $[\rightarrow, Set]$-*Cat*.

4. Premonoidal categories

In this section, we recall the definitions of premonoidal category and symmetric premonoidal category, and outline how mild variants of them may be seen as algebraic structures on $[\rightarrow, Set]$-*Cat* and *Subset*-*Cat* (see [16] for the definitions in fuller form and for examples).

Definition 10. A *binoidal category* is a category C together with, for each object x of C, functors $h_x : C \rightarrow C$ and $k_x : C \rightarrow C$ 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 11. An arrow $f : x \rightarrow y$ in a binoidal category is *central* if for every arrow $g : u \rightarrow v$, the two composites from $x \otimes u$ to $y \otimes v$, one given by $k_v(f) \cdot h_x(g)$, the other given by $h_y(g) \cdot k_u(f)$, are equal, and dually for the two composites from $u \otimes x$ to $v \otimes y$.

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 $f : x \rightarrow y$ is central and z is any object of C, then $h_z(f) : z \otimes x \rightarrow z \otimes y$ and $k_z(f) : x \otimes z \rightarrow y \otimes z$ are central.

Definition 12. Given a binoidal category C, a natural transformation $\alpha : g \Rightarrow h : B \rightarrow C$ is called *central* if every component of α is central.

Definition 13. A *premonoidal category* is a binoidal category C together with an object I of C, and central natural isomorphisms a with components $(x \otimes y) \otimes z \to x \otimes (y \otimes z)$, l with components $x \to x \otimes I$, and r with components $x \to I \otimes x$, 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. So, trivially we may extend our study of Example 4 by observing

Example 14. For any elementary control structure $j : M \to D_0$, the category D_0 is a premonoidal category.

For a more substantial example, extending Example 5, we have

Example 15. The category CBV is a premonoidal category. The easiest way to verify that is by direct use of the definition of premonoidal category.

Finally, we may extend Example 6 to observe that, by definition,

Example 16. Every $\otimes\neg$-category is a 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. It is needed to characterize strong monads and their various conditions in this setting.

Definition 17. 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, as we have in the definition of premonoidal category, h_z and k_z preserve central maps. It immediately follows that we have

Proposition 18. *The centre of a premonoidal category is a monoidal category.*

The notion of the centre of a premonoidal category was vital in the development of $\otimes\neg$-categories as in Example 6 (see [17, 20, 19]). The central morphisms of a premonoidal category provided a notion of effect-free term or value in the study of continuation semantics. An analysis of centrality as defined here gave rise to the refutation of several conjectures about control effects [19].

One of the central themes of this paper is to identify, as part of the structure, a specified subcategory of the centre of a premonoidal category; that allows us to prove Theorem 22. But this is one case in which the consideration of all central maps, not just some of them, is vital, because it is precisely the fact of centrality that has provided the notion of effect-free term.

For Example 15, CBV_t lies in the centre of CBV.

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

Definition 19. A *symmetry* for a premonoidal category is a central natural isomorphism with components $c: x \otimes y \to y \otimes x$, 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.

All our leading examples of premonoidal categories are symmetric. We now define the notion of premonoidal functor. This generalises the notion of monoidal functor (see [2]).

Definition 20. A *premonoidal functor* $(g, \bar{g}, \hat{g}): C \to D$ is a functor $g: C \to D$ that sends central maps to central maps, together with central natural transformations with components $\bar{g}: ga \otimes gb \to g(a \otimes b)$ and $\hat{g}: I \to g(I)$, subject to the three equations expressing coherence with a, l and r. A premonoidal functor is called *strong* or *strict* when $\bar{g}$ and $\hat{g}$ are isomorphisms or identities, respectively.

One may similarly generalise the definition of symmetric monoidal functor to symmetric premonoidal functor. 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. We will denote the 2-category of small premonoidal categories, strict premonoidal functors, and premonoidal natural transformations by $Premon_s$.

To see why the forgetful functor $U: Premon_s \to Cat$ does not present the former as being of the form (S,E)-Alg for algebraic structure on Cat, we appeal to Theorem 7: if it did, then it would be the forgetful functor from a category of algebras T-Alg to Cat. So it would have equalisers given as in Cat. However, every monoid may be seen as a one object premonoidal category. So consider

Example 21. Let $\{a,b\}^*$ be the free (noncommutative) monoid on two elements, and consider the two maps from $\{a,b\}^*$ to itself given by the identity map and by the map sending a to itself and b to the unit. These are both strict premonoidal functors. Their equaliser in Cat is given by $\{a\}^*$, the free monoid on the element a. However, the centre of $\{a\}^*$ is itself, as every element of $\{a\}^*$ commutes with every other element. But the inclusion of $\{a\}^*$ into $\{a,b\}^*$ does not preserve centrality, as the centre of $\{a,b\}^*$ is merely the unit. Hence, the inclusion is not a strict premonoidal functor, and hence the equalizer does not lift.

One might wonder whether the functor from $Premoncat_s$ to Cat that sends a premonoidal category to its centre may be monadic; but again, it is not: all monadic functors are conservative, i.e., reflect isomorphisms, and this one is not, because a strict premonoidal functor that restricts to an isomorphism between centres need not be an isomorphism, as for instance in the second map in Example 21.

So it seems impossible to treat premonoidal structure as algebraic structure on *Cat*, and we need something more subtle to reconcile premonoidal structure with our usual account of structured categories. As indicated earlier, we turn to algebraic structure on $[\rightarrow, Set]$-*Cat* and *Subset*-*Cat* to do so. That is the content of the next section.

5. Premonoidal categories as *Subset*-categories with algebraic structure

There is little difference between *Subset*-categories and $[\rightarrow, Set]$-categories, and it is not clear yet which provide the better setting for our analysis. Trivially, every *Subset*-category is a $[\rightarrow, Set]$-category, but not conversely. Two of our leading examples, Examples 5 and 6, are *Subset*-categories, but Example 4 is not. Moggi first opted for something equivalent to a special case of *Subset*-categories in defining a computational model (which he defined to be a monad satisfying the mono requirement), and later moved towards $[\rightarrow, Set]$-categories, which provide the models for his computational lambda calculus [10]. In terms of category theory, the latter are a little more general and abstractly a little more natural; but the former are a little easier to handle concretely. The results we present here hold for both, with essentially the same proofs. So we will present them only for $[\rightarrow, Set]$-categories to avoid clutter.

In order to obtain a coherence result for premonoidal categories, one needed to assert that all structural isomorphisms were central. Then in order for small premonoidal categories and premonoidal functors to form a category, one needed to assert that premonoidal functors send central maps to central maps. So this gives us a hint that we may be better not to attempt to treat a premonoidal category as a single category with algebraic structure, but rather to consider a category, together with another category and an identity on objects functor between them, and consider algebraic structure on that. To give a pair of categories and an identity on objects functor between them is precisely to give a $[\rightarrow, Set]$-category. Thus we are led to consider $[\rightarrow, Set]$-categories with algebraic structure.

When one does that, everything falls into place remarkably well. For instance, recall from Section 2 that to give a $[\rightarrow, Set]$-functor is to give a commutative square of functors, and that is the condition required of a premonoidal functor with respect to its centre. Moreover, to give a $[\rightarrow, Set]$-natural transformation is to give a natural transformation between the functors between the codomain categories, but with components in the domain category, and that condition corresponds to that of central natural transformations. In fact, we have

Theorem 22. *There is algebraic structure on* $[\rightarrow, Set]$-*Cat for which an algebra is a small premonoidal category D together with a monoidal category C and a strict premonoidal identity on objects functor* $j : C \rightarrow D$.

Proof. Let $\rightarrow$ denote the $[\rightarrow, Set]$-category with two objects, with one arrow from the first to the second in the codomain category, and with the domain category discrete.

Let $\to_c$ denote the $[\to, Set]$-category with two objects and an arrow from one to the other in the codomain category and the domain category. Let 1 denote the discrete $[\to, Set]$-category with one object.

Let $j: C \to D$ be an arbitrary small $[\to, Set]$-category. Then the category $[\to, Set]$-$Cat(1, j)$ is isomorphic to C. Also, an object of the category $[\to, Set]$-$Cat(\to, j)$ is an arrow of D, and an arrow is a pair of arrows in C that together with the domain and codomain, form a commutative square in D. The category $[\to, Set]$-$Cat(\to_c, j)$ is the full subcategory of $[\to, Set]$-$Cat(\to, j)$ given by the arrows of C.

So if we put

- $S(1 + \to) = \to$,
- $S(\to + 1) = \to$, and
- $S(c) = 0$ for all other c,

then an S-algebra would consist of a $[\to, Set]$-category $j: C \to D$, together with the data for functors $h_x: D \to D$ and $k_x: D \to D$ for each object x, with a little more data and naturality conditions that may be used force each map in C to be sent by j to a central map. One can extend S by operations and equations to force the above data to give D the structure of a binoidal category: one needs to ensure that the object functions of the two functors are well defined and agree as required by the binoidal definition, and that composition and identities are preserved. In doing so, $J: C \to D$ is forced to factor through $Z(D)$. Then one can routinely add operations and equations to give the coherent structural isomorphisms a, l, and r, making D premonoidal. □

By Theorem 22, the 2-category given by

- an object is a small premonoidal category D together with a monoidal category C and a strict premonoidal identity on objects functor $j: C \to D$,
- an arrow is commutative square in *Cat* that strictly respects the tensor product and structural isomorphisms,
- a 2-cell is a natural transformation between the functors between the codomain categories, with components in the domain category, strictly respecting the tensor product and the structural isomorphisms.

$[\to, Set]$-Cat. We denote this 2-category by $[\to, Set]$-$Premon_s$ and call it the 2-category of premonoidal $[\to, Set]$-categories and strict premonoidal $[\to, Set]$-functors. Note that an arrow need not be a strict premonoidal functor in general, as we do not assert that the functor preserves centrality: we merely assert that it respects the domain category structure. So if there is a central map of the domain $[\to, Set]$-category that is not in the image of the domain category, we do not assert that it need be sent to a central map. However, it is routine to verify that by forcing the domain category to be the centre, we do have

Proposition 23. *The 2-category of strict premonoidal categories*, $Premon_s$, *embeds fully into* $[\to, Set]$-$Premon_s$.

One can account for symmetry in this setting similarly. Moreover, as mentioned in Section 2, one can extend any algebraic structure on *Cat* to algebraic structure on $[\rightarrow, Set]$-Cat: you just apply a diagonal to the ordinary structured category to obtain a corresponding structured $[\rightarrow, Set]$-category.

These results suggest that one might consider $[\rightarrow, Set]$-categories as worthy of substantial investigation as a way of organizing many of the structures involved with semantics of computation. Since $[\rightarrow, Set]$ is a locally finitely presentable cartesian closed category, one immediately may use the results of enriched category theory. For instance, $[\rightarrow, Set]$ may itself be treated as a $[\rightarrow, Set]$-category, and plays much the same role as that of *Set* in ordinary category theory, so one can speak of presheaves, free cocompletions, etcetera. Moreover, $[\rightarrow, Set]$-Cat is a locally finitely presentable 2-category, so one immediately has access to the whole body of literature about 2-monads, as outlined in [12], and in particular the treatment of functors that preserve structure only up to coherent isomorphism, as in [2]. In particular, this gives us the lax and strong notions of structure preservation for algebraic structure, and they agree, modulo the embedding, with the definitions of premonoidal functor and strong premonoidal functor.

More specific to our concerns here, $[\rightarrow, Set]$-Cat may be treated as an ordinary category rather than a 2-category, and one may consider algebraic structure on it as an ordinary category. If we do that, then it follows from Example 9 that we have

Example 24. The category of small $[\rightarrow, Set]$-categories $j : C \rightarrow D$ for which C has (strictly associative) finite products, D is (strict) symmetric monoidal, and j is strict symmetric monoidal, is given by algebraic structure on $[\rightarrow, Set]$-Cat. For the non-strict case, the proof is given by extending that of Theorem 22: take the structure for Theorem 22, extend the definition of S to provide central diagonal and terminal maps for each object and a central symmetry map for each pair of objects, then add equations for the coherence, and add more equations to make D monoidal rather than premonoidal and for all the necessary naturality (cf. [8]). For the strict case, the algebraic structure required for a strict version of Theorem 22 is considerably simplified, just as algebraic structure for strict monoidal categories is simple relative to that for monoidal categories [8]. If one consistently accounts for that, then the proof again extends routinely.

So this accounts for all the type structure in the definition of elementary control structures. Since Example 5 is an example of a specific category rather than an example of a class of categories with structure, we cannot make a similar statement about it. But we can extend Example 6 to yield

Example 25. The category of small premonoidal $[\rightarrow, Set]$-categories $j : C \rightarrow D$ together with an endofunctor $\neg : D^{op} \rightarrow C$ satisfying all the axioms of $\otimes\neg$-categories except for the demand that D comprise all the central maps, is given by algebraic structure on $[\rightarrow, Set]$-Cat. A proof of this again extends that of Theorem 22. Extend S to provide the data for the endofunctor $\neg : D^{op} \rightarrow C$, then add equations for functoriality and for all the axioms, all of which are equational in the precise sense of this section.

6. Closed premonoidal categories

We now seek an account of closedness for a premonoidal $[\rightarrow, Set]$-category. There are two definitions that immediately generalise the definition of closed monoidal category. One of them just extends the usual bijection in the definition of monoidal category to a premonoidal category, with appropriate coherence. We do not yet see its value in modelling higher order structure, so do not investigate it here. The other is equivalent to that used by Moggi in [10], generalising the partial exponential one uses in modelling partial functions. We investigate that in this section.

Definition 26. A premonoidal $[\rightarrow, Set]$-category $j : C \rightarrow D$ is *closed* if for every object x, the functor $j(-) \otimes x : C \rightarrow D$ has a right adjoint.

Proposition 27. *Let $j : C \rightarrow D$ be a premonoidal $[\rightarrow, Set]$-category such that C is closed monoidal. Then j is closed if and only if j has a right adjoint.*

Proof. Assuming j is closed, take x to be the unit to see that j has a right adjoint. For the converse, compose the right adjoint with the closed structure of C. □

For a class of examples of closed symmetric premonoidal $[\rightarrow, Set]$-categories, we appeal to the main result of [16]. Given a monad T on a category C, we denote the Kleisli category by K_T. Observe that the canonical functor from C to K_T is the identity on objects.

Extending our leading examples,

Example 28. In general, an elementary control structure does not yield a closed premonoidal category. However, those elementary control structures that are closed form a natural class to analyse, as has been done in Hasegawa's thesis [5].

Example 29. The inclusion of CBV_t into CBV is a closed symmetric premonoidal category.

Example 30. It follows from the axioms for a $\otimes\neg$-category that the inclusion of the centre of a $\otimes\neg$-category into the whole of the $\otimes\neg$-category is a closed symmetric premonoidal category.

Returning to the general development of this section,

Theorem 31. *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.*

Proposition 27 and Theorem 31 together imply

Corollary 32. *Let C be a closed symmetric monoidal category. Then, to give a closed symmetric premonoidal* $[\rightarrow, Set]$*-category with domain C is to give a strong monad on C.*

Considering Examples 29 and 30 in light of Corollary 32, one might ask whether one really needs the notions developed here, or whether one could use strong monads instead. However, in the literature [20], a $\otimes\neg$-category was *defined* to be a premonoidal category with extra structure, and its use directly reflected its definition, with the premonoidal structure used to model contexts in a call-by-value λ-calculus with control. From the extra structure, one can deduce that the underlying premonoidal category of a $\otimes\neg$-category is closed. But that does not imply that one can reasonably redefine the notion of $\otimes\neg$-category in terms of strong monads: you still have to find a way to express the full strength of the extra structure and its axioms, and those axioms refer directly to the premonoidal structure, and one must do this without redundancy. It is not clear that that is possible, and even if it were, it would not be in the spirit of providing direct models of languages with control as Thielecke sought [20].

The main result of this section shows how to generate and explicitly describe a closed premonoidal $[\rightarrow, Set]$-category from an arbitrary small premonoidal $[\rightarrow, Set]$-category. This generalises the result [4] and explicit construction of Brian Day, that shows that every small monoidal category embeds into a closed monoidal category. The result and proof here extends without any difficulty to a version for symmetric premonoidal *Subset*-categories too.

Theorem 33. *Every premonoidal* $[\rightarrow, Set]$*-category fully embeds into a closed premonoidal* $[\rightarrow, Set]$*-category.*

Proof. Take the free cocompletions of C and D, yielding $[C^{op}, Set] \rightarrow [D^{op}, Set]$ with a right adjoint. We will define C' to be $[C^{op}, Set]$ and define D' and j' by the (bijective on objects, fully faithful) factorization of this functor. Then, j' automatically has a right adjoint, and C' is automatically closed monoidal. The only remaining problem is to get a premonoidal structure on D' so that j' factors through $Z(D)$.

Given any object x of D, then by the universal property, $x \otimes - : C \rightarrow D$ lifts to C', and has a right adjoint there. Moreover, for any arrow $g : x \rightarrow x'$ in D, g yields a natural transformation $g \otimes - : x \otimes - => x' \otimes - : C \rightarrow D$, and hence by universality (the two-dimensional property), a natural transformation $g \otimes -$ on C'. This construction of a natural transformation is functorial, yielding $- \otimes F : D \rightarrow [D^{op}, Set]$ for each F in C'. By universality again, this lifts to $[D^{op}, Set]$ and hence to D'. This defines $- \otimes F$ on D' for any object F of D'. It is by definition a functor. By symmetry, we also have $F \otimes -$. It follows from the universal constructions that it extends the tensor on C'. It remains to show that j' factors through $Z(D')$.

It suffices to show that for any $f : F \rightarrow F'$ in C', $f \otimes -$ is natural on D'. To do that, by the universal property, it suffices to see that $f \otimes -$ is natural on D, but that holds by definition of $F \otimes g$ for any g in D (the dual is given above) as it is defined to be natural with respect to F. □

Note that the construction gives us a little more than in the statement of the theorem: our construction has C' closed monoidal.

Corollary 32, Theorem 33, and Definition 3 suggest a new structure to consider to model notions of computation: a premonoidal $[\rightarrow, Set]$-category $j: C \rightarrow D$ with C cartesian, together with, for a given set of controls K with arity information $((m_1,n_1), \ldots, (m_r,n_r)) \mapsto (m,n)$, a corresponding set of natural transformations $D(j(-) \times m_1, n_1) \times \cdots \times D(j(-) \times m_r, n_r) \rightarrow D(j(-) \times m, n)$.

Using naturality, it is routine to extend Theorem 33 to conclude

Corollary 34. *Every structure as above fully embeds into a closed premonoidal* $[\rightarrow, Set]$*-category* $j': C' \rightarrow D'$ *with* C' *cartesian, together with, for each control K with arity information* $((m_1,n_1), \ldots, (m_r,n_r)) \mapsto (m,n)$, *an arrow in* C' *of the form* $(m_1 \Rightarrow n_1) \times \cdots \times (m_r \Rightarrow n_r) \rightarrow (m \Rightarrow n)$, *where* $m \Rightarrow n$ *is defined by the exponential of* j', *i.e.*, $[m, R(n)]$, *where R is the right adjoint of* j'.

Proof. That C' is cartesian closed follows from its construction in Theorem 33. For the controls, D is a full subcategory of D', so the control natural transformations may be re-expressed as natural transformations of the form $D'(j'y(-) \times m_1, n_1) \times \cdots \times D'(j'y(-) \times m_r, n_r) \rightarrow D'(j'y(-) \times m, n)$, where y is the Yoneda embedding of C into $C' = [C^{op}, Set]$. But j' has a right adjoint R, and representable functors preserve finite products, so this in turn is equivalent to giving a natural transformation of the form $C'(y(-), (m_1 \Rightarrow R(n_1)) \times \cdots \times (m_r \Rightarrow R(n_r))) \rightarrow C'(y(-), m \Rightarrow n)$, which can be proved equivalent, either by calculation using the Yoneda lemma or by reference to the fact that C is dense in C', to giving an arrow as stated. □

There are plenty of examples of these structures. The structures of Corollary 34 amount to strong monads together with some operators, giving operator algebras. Categories such as *Set* and the category of ω-cpo's provide commonly used base categories, and premonoidal categories arise in nondeterminism for example, for instance in [1], where the operators are particularly vivid.

Recalling the work of the previous section, we note

Example 35. The category of small closed premonoidal categories is given by unenriched algebraic structure on $[\rightarrow, Set]$-*Cat*. For a proof, consider the proof that the category of small monoidal closed categories is given by algebraic structure on *Cat*, and modify the proof for closedness to premonoidal categories by extending the construction of Theorem 22.

Finally in this section, we mention that if C is closed symmetric monoidal, in particular, cartesian closed, to give a strength to a monad on C is to give an enrichment; but typically, as in *Poset* and the category of ω-cpo's with least element, there is at most one enrichment of any monad. So there is often at most one closed symmetric premonoidal structure on a $[\rightarrow, Set]$-category that extends the structure of the domain category. So proving that a category has one closed symmetric premonoidal structure is a strong statement: there may well be no others.

7. Applications and future work

The work herein is part of a larger project to give a modular account of denotational semantics. It was proposed that strong monads would provide such an account, but it was shown in [14] that there must be a more primitive structure that gives rise to monads. The proposal in [14] was to consider a category C with algebraic structure, precisely as defined herein, together with a construction from that algebraic structure of another category D and an identity on objects functor $j : C \rightarrow D$, i.e., a $[\rightarrow, Set]$-category. That was supported by a theorem showing when one such structure was modular with respect to another.

There is more work to do in that direction. The notion of algebraic structure was made precise in [14], as was the idea of how to lift one structure along another; but it was not made precise what constructions of $[\rightarrow, Set]$-categories were allowable. Moreover, the analysis was done for structure on *Cat*, so excluded premonoidal structure as arises for instance in modelling continuations or state. So we should like to combine the work of that paper with this one in order to provide modular denotational semantics.

Moreover, the results of [14] were restricted to covariant structures, thus excluding higher order structure. This paper gives a definitive notion of closed premonoidal category, so provides a basis on which to analyse closed structures and to incorporate them into the proposed modular semantics.

The structures developed herein, namely *Subset*-categories and $[\rightarrow, Set]$-categories with algebraic structure, form part of that structure, addressing the question of how to model contexts. These structures have already been used in modelling continuations. There is current work by Alan Jeffrey using them for circuit design, and by Carsten Fuhrman, using them to model other controls, and by Paul Levy as a semantic environment in which to study the CPS-calculus [20]. The results here give a criterion for deciding what added structures are mathematically respectable in addressing the particular concerns these researchers face.

Finally, in [17], $[\rightarrow, Set]$-categories have been characterized in terms of fibrations with structure, and that characterization has been and continues to be used to give further analysis of continuations, shedding light on the complex axioms associated with the definition of $\otimes\neg$-category, specifically a parametric aspect of the axioms.

References

[1] S.O. Anderson, A.J. Power, A representable approach to finite nondeterminism, Proc. MFPS '94, Theoret. Comput. Sci. 177 (1997) 3–25.

[2] R. Blackwell, G.M. Kelly, A.J. Power, Two-dimensional monad theory, J. Pure Appl. Algebra 59 (1989) 1–41.
[3] M. Fiore, Axiomatic domain theory in categories of partial maps, Cambridge University Press Distinguished Dissertations in Computer Science, Cambridge, 1996.
[4] Guen Bin Im, G.M. Kelly, A universal property of the convolution monoidal structure, J. Pure Appl. Algebra 43 (1986) 75–88.
[5] M. Hasegawa, Models of sharing graphs: a categorical semantics of let and letrec, Edinburgh University Thesis, 1997, ECS-LFCS-97-360 (to appear as a Distinguished Dissertation in Computer Science).
[6] K. Honda, N. Yoshida, Game theoretic analysis of call-by-value computation, Proc. ICALP 97, Lecture Notes in Computer Science, Vol. 1256, Springer, Berlin, 1997, pp. 225–236.
[7] G.M. Kelly, in: Basic concepts of enriched category theory, London Mathematical Society Lecture Notes Series, Vol. 64, Cambridge University Press, Cambridge, 1982.
[8] G.M. Kelly, A.J. Power, Adjunctions whose counits are coequalizers, and presentations of finitary enriched monads, J. Pure Appl. Algebra 89 (1993) 163–179.
[9] S. Mac Lane, Categories for the Working Mathematician, Springer, Berlin, 1971.
[10] E. Moggi, Notions of computation and monads, Inform. and Comput. 93 (1991) 55–92.
[11] A. Mifsud, R. Milner, A.J. Power, Control structures, Proc. LICS 95 (1995) 188–198.
[12] A.J. Power, Why tricategories?, Inform. Comput. 120 (1995) 251–262.
[13] A.J. Power, Elementary control structures, Proc. CONCUR '96: Concurrency Theory, Lecture Notes in Computer Science, 1996, Springer, Berlin, pp. 115–130.
[14] A.J. Power, in: Modularity in Denotational Semantics, Proc. MFPS 97, Electronic Notes in Theoretical Computer Science 6 (1997).
[15] A.J. Power, E.P. Robinson, On the categorical semantics of types, Draft, 1995.
[16] A.J. Power, E.P. Robinson, Premonoidal categories and notions of computation, Proc. LDPL '96, Math Struct. Comput. Sci. 7 (1997) 453–468.
[17] A.J. Power, H. Thielecke, in: Environments, continuation semantics, and indexed categories, Proc. Theoretical Aspects of Computer Science, Lecture Notes in Computer Science, Vol. 1281, Springer, Berlin, 1997.
[18] E. Robinson, Variations on algebra: monadicity and generalisations of algebraic theories, Math. Struct. Comput. Sci., to appear.
[19] H. Thielecke, Continuation passing style and self-adjointness, Proc. 2nd ACM SIGPLAN Workshop on Continuations, BRICS Notes Series NS-96-13, 1996.
[20] H. Thielecke, in: Continuation semantics and self adjointness, Proc. MFPS 97, Electronic Notes in Theoretical Computer Science 6 (1997).