Symmetric monoidal sketches* **

Martin Hyland[1] and John Power[2]

[1] Department of Pure Mathematics and Mathematical Statistics, University of Cambridge, Mill Lane, Cambridge CB2 1SB, England
email: M.Hyland@dpmms.cam.ac.uk Tel: +44 1223 337 986
[2] Laboratory for the Foundations of Computer Science, University of Edinburgh, King's Buildings, Edinburgh EH9 3JZ, Scotland
email: ajp@dcs.ed.ac.uk Tel: +44 131 650 5159 Fax: +44 131 667 7209

Abstract. We introduce and develop the notion of symmetric monoidal sketch. Every symmetric monoidal sketch generates a generic model. If the sketch is commutative and single-sorted, the generic model can be characterised as a free structure on 1, and the construction sending a small symmetric monoidal category to the category of models of the sketch in it can be seen as a right adjoint. We investigate specific cases generated by the Eckmann-Hilton argument, which allows a simple characterisation of the constructions. This accounts for the various categories of wiring currently being investigated in modelling concurrency, as well as providing a basis for understanding the axiomatically generated categories in axiomatic domain theory and in presheaf models of concurrency.

1 Introduction

In recent years, in the studies of concurrency [11, 12, 3] and denotational semantics [5], as part of a broad attempt to give an axiomatic account of programming language semantics, it has been common to consider a free symmetric monoidal category with some extra data or some extra axioms on 1. For instance, Milner's category of wirings [11, 12] is given by the free symmetric monoidal category for which the symmetric monoidal structure is finite product structure on 1; Fiore et al's cuboidal sets [5] are defined by considering a free symmetric monoidal category with a lifting monad, subject to additional axioms, on 1; and Winskel et al's work on presheaf models for concurrency [3] requires "path" categories, which are given by free symmetric monoidal categories with additional structure on 1. Currently, Plotkin is developing a semantics for CCS, and he too considers a free symmetric monoidal category, subject to some axioms, on 1 in order to give a wiring category, a different wiring category to that of Milner. Philippa Gardner, Alex Simpson and others are doing likewise, and it seems

* This work has been done with the support of EPSRC grant GR/M56333 and a British Council grant,and the COE budget of STA Japan.
** **Keywords:** symmetric monoidal category, sketch, action calculi, categories of wirings

likely that this trend will continue. So we seek a solid mathematical foundation to support these endeavours.

A first point to note is that one can characterise all of these constructions in different terms. For instance, one can construct Milner's category of wirings as follows: consider symmetric monoidal structure and take a commutative monoid; it consists of an object X together with maps $j : I \longrightarrow X$ and $m : X \otimes X \longrightarrow X$ subject to the commutativity of four diagrams, one each for left unit, right unit, associativity, and commutativity. Now consider the free symmetric monoidal category generated by such a commutative monoid: it is equivalent to the category Set_f of finite sets and functions, which in turn is equivalent to Milner's construction.

There is one difficulty with this analysis and that is that formally, it is not mathematically precise in that we have not said exactly what we mean by "consider symmetric monoidal structure". So in this paper, we make the idea of the above definition and construction precise, including examples such as commutative monoids, commutative comonoids, bimonoids, and relational bimonoids, and see what results follow in general, rather than having to deal with the specific examples separately. We restrict our attention to symmetric monoidal structure, not dealing with the additional structure such as that of a lifting monad, but our results extend to "pseudo-commutative 2-monads on Cat," allowing us to include endofunctors and the like, which we shall do later.

The central definitions we develop here are those of *symmetric monoidal sketch* S and the category of strict models $Mod_s(S, C)$ (in Section 2) of S in any symmetric monoidal category C. In particular, any symmetric monoidal sketch S generates a *generic* model or *theory*, $Th(S)$. The generic model is characterised by the property that if C is a small symmetric monoidal category, $Mod_s(S, C)$ is isomorphic to the category $SM_s(Th(S), C)$ of strict symmetric monoidal functors from $Th(S)$ to C, and this is natural in C. Such generic models are exactly the various categories of wirings, etcetera, studied by Milner et al, and, subject to the addition of a little more structure, by Fiore, Winskel, et al.

We say S is single-sorted (see Section 3 for details) if all the objects of S are generated by a single base object X: this holds in all our examples, such as that above for a commutative monoid. We say S is commutative if all the maps in S commute with each other in a precise sense. If S is both commutative and single-sorted, it follows that every object of $Th(S)$ has an S-structure on it, so for instance, every object of $Th(CMon)$, where $CMon$ is the sketch for a commutative monoid, has a commutative monoid structure. That allows us to characterise $Th(S)$ as the free category with structure on 1, which is how several authors have presented their work, e.g., Fiore et al [5] and Milner et al [11, 12].

For any commutative single-sorted sketch S, the functor $Mod_s(S, -)$ possesses the structure of a comonad, so the construction that sends a small symmetric monoidal category C to the category $Mod_s(S, C)$ of strict models of S in C can be characterised as a right adjoint. For many sketches, including those of primary interest to us, there is an elegant characterisation of the target category, $Mod_s(S, -)$-$Coalg$, of that right adjoint. We explore the situation in Section 4.

The reason is essentially the Eckmann-Hilton argument [4]: this shows that given a group in the category of groups, the two group structures must agree and must be Abelian. This result generalises to all our leading examples, showing that $Mod_s(S, Mod_s(S, C))$ is coherently isomorphic to $Mod_s(S, C)$, which in turn allows us to characterise $Mod_s(S, -)$-$Coalg$. For instance, in our commutative monoid example, this shows that the forgetful functor from the category of small categories with finite products to the category of small symmetric monoidal categories has a right adjoint given by sending a small symmetric monoidal category C to the category $CMon(C)$ of commutative monoids in C.

The main mathematical technique we use in the course of the paper is the theory of sketches for an arbitrary finitary 2-monad on Cat as developed in [10]. The reader does not require knowledge of that work to follow this paper. For most of the paper, for ease of exposition, we shall gloss over coherence questions relating to the distinction between preservation and strict preservation of category theoretic structure: part of the reason that does not create major difficulty is because every monoidal category is equivalent to a strict monoidal category, so we shall tend to conflate the two notions: only at one point, where we define single-sortedness, might that be a little misleadingly simple, but we have been careful to be correct. We investigate the relevant two-dimensional issues seriously in Section 5. Ultimately, they may be resolved by reference to [2] and [9].

While writing this paper, we have also written [7] using some of the mathematical techniques developed here but specifically directed towards generalising Milner's wiring category. That paper contains more detailed examples, but it does not contain any two-dimensional analysis, it has less focus on commutative sketches, and its only examples are those of wiring categories as developed for concurrency.

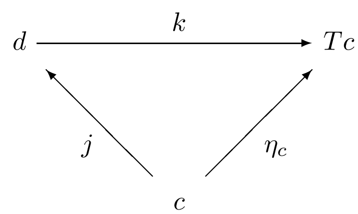
The paper is organised as follows. In Section 2, we define the notion of symmetric monoidal sketch, the category $Mod_s(S, C)$ of strict models of a sketch S in an arbitrary small symmetric monoidal category C and the notion of generic model $Th(S)$ of a symmetric monoidal sketch S, and we give our leading examples. In Section 3, we investigate general conditions under which $Th(S)$ can be seen as the free category with specified structure generated by 1. This involves defining the notions of single-sortedness and commutativity of a sketch. In Section 4, we generalise the Eckmann-Hilton argument, for all our leading examples, to characterise the target category $Mod_s(S, -)$-$Coalg$ of the right-adjoint functor $Mod_s(S, -)$. Finally, in Section 5, we explain the subtleties that arise in needing to distinguish between preservation and strict preservation of category theoretic structure, and how they can be resolved in this context.

2 The definition of a symmetric monoidal sketch

There has long been study of finite product sketches [1]. But one needs some subtlety in adapting that definition to symmetric monoidal structure: symmetric monoidal structure does not have cones, but it does have non-identity structural maps, such as the symmetry maps as in the commutativity diagram for a commutative monoid, that must be respected, and no coherence theorem can avoid that. So our definition requires care.

We first need to define the notion of a family of diagram types. This is unnecessary in defining finite product sketches as all the properties of products are determined by their universal property, and the data required for the universal property is completely given by universal cones. For symmetric monoidal categories, that is not the case. Despite this being the first definition of the paper, it is ultimately a supplementary definition, as we shall soon see.

2.1 Definition A *family D of diagram types* is a small family of 4-tuples $(c_i,\ d_i,\ j_i\colon c_i \to d_i,\ k_i\colon d_i \to Tc_i)$, where c_i and d_i are finitely presentable categories, Tc_i is the free symmetric monoidal category on c_i, and j_i and k_i are functors, subject to the condition that the following diagram, dropping the subscripts, commutes:



where η is the unit of T.

We generally suppress j and k, leaving them implicit in c and d. So we speak of (c, d). An example follows our definition of sketch.

2.2 Definition A *symmetric monoidal sketch* S consists of a small category X together with a a family D of diagram types and a D-indexed family of functors $\phi_i : d_i \longrightarrow X$. A map of symmetric monoidal sketches from (X, ϕ_i) to (Y, ψ_i) is a functor $f : X \to Y$ such that $f\phi_i = \psi_i$ for each i.

For fixed D, sketches and maps of sketches give a category $SymMonD$-$Sketch$, which is finitarily monadic over $Cat_\circ$: in fact, we can enrich it to give a 2-category finitarily monadic over Cat using structure-respecting natural transformations: this observation becomes crucial when we investigate coherence issues in Section 5.

We now turn to the notion of a model of a sketch. A sketch S has models in a small symmetric monoidal category. We use the term strict model here: strict models relate to models as functors sending assigned finite products are to functors that preserve finite products in the usual sense. This has been made precise in [2] and [9]. The former are easier for writing an abstract account, but the latter are more natural. So for most of the paper, we restrict attention to the former, but in Section 5, we shall explain the more refined notion of model and how a theory of models follows from the theory we present of strict models.

Let C be a small symmetric monoidal category, and let $S = (X, \phi)$ be a symmetric monoidal sketch: we drop the subscripts on the elements of the family D as they are clear.

2.3 Definition A *strict model* of (X, ϕ) in C is a functor $f : X \longrightarrow C$ such that the following diagram commutes:

$$\begin{array}{ccc} d & \xrightarrow{\ k\ } & Tc \\ {\scriptstyle \phi}\Big\downarrow & & \Big\downarrow{\scriptstyle (f\phi j)^*} \\ X & \xrightarrow[\ f\ ]{} & C \end{array}$$

where $(f\phi j)^*$ is given by using freeness of Tc.

For most of the paper, we shall simply refer to these as models rather than strict models.

For small symmetric monoidal categories B and C, there is a homcategory $SM_s(B, C)$ as usual. For a symmetric monoidal sketch $S = (X, \phi)$ and a small symmetric monoidal category C, we need to make the set of strict models of S in C into a category.

2.4 Definition The object $Mod_s(S, C)$ is defined to be the limit in Cat of the diagram with vertex $Cat(X, C)$ and for each ϕ_i, two maps from $Cat(X, C)$ to $Cat(d_i, C)$, the first given by composition with ϕ_i, the second given by first precomposing with $\phi_i j_i$, then applying $(\,)^*$, then precomposing with $k_i : d_i \longrightarrow Tc_i$.

The central result of [10] yields

2.5 Theorem For any symmetric monoidal sketch S, there is a small symmetric monoidal category $Th(S)$ and there is a model ι of S in $Th(S)$ such that composition with ι induces an isomorphism of categories from $SM_s(Th(S), C)$ to $Mod_s(S, C)$.

We call $Th(S)$ together with $\iota : S \longrightarrow Th(S)$ the *generic* model of S.

For examples, we start with something that illustrates how the definitions work but is not of itself of interest to us. We then proceed with our leading examples.

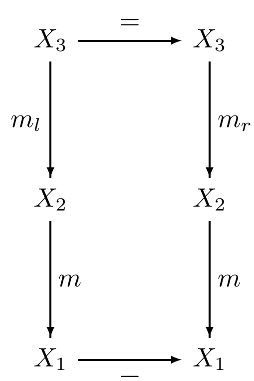
2.6 Example Let D consist of one pair $(2, 3)$, with j the (ordered) inclusion of 2 into the first two components of 3, and k the inclusion of 3 into $T(2)$ yielding that part of $T(2)$ that gives the tensor product of the two base objects. That it satisfies the condition on a family of diagram types amounts to the assertion that k sends the first two components of 3 to the respective generating objects of $T(2)$.

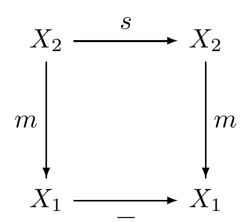
A symmetric monoidal sketch $\mathcal{S}$ with D as above is a small category, which we also denote by S, together with a sequence of three objects (c, d, e). It follows from our definition of model of a sketch that we could reasonably denote the object e by $c \otimes d$.

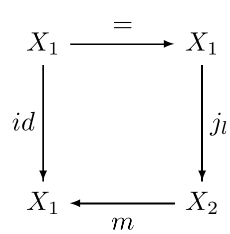
A model of S in a small symmetric monoidal category C is a functor $H : S \longrightarrow C$ such that $H(e) = H(c) \otimes H(d)$, This determines a strict symmetric monoidal functor from $Th(S)$ to C. Since it preserves symmetric monoidal structure strictly and it extends H, it still takes e to $H(e) = H(c) \otimes H(d)$.

2.7 Example Let $CMon$ be the sketch for a commutative monoid. As we mentioned before, every monoidal category is equivalent to a strict monoidal category, so for general category theoretic reasons [2], it is safe for us to conflate the two notions. So in describing the sketch here, we shall do that for simplicity.

The underlying category of $CMon$ is that required to express the data and commutativity axioms for a commutative monoid: so it has four objects X_0, X_1, X_2, and X_3. Its arrows are freely generated by arrows $j : X_0 \longrightarrow X_1$, $m : X_2 \longrightarrow X_1$, $m_l, m_r : X_3 \longrightarrow X_2$, $s : X_2 \longrightarrow X_2$, and $j_l : X_1 \longrightarrow X_2$, subject to commutativity of the following diagrams:







The family D and the maps $\phi_i : d_i \longrightarrow M$ are those required to force X_0 to be sent to the unit in any model, X_2 to be sent to the tensor product of the image of X_1 with itself in any model, etcetera. So the sketch has two diagram types. The first, (c_0, d_0, j_0, k_0), has c_0 as the category with one object A and no non-trivial arrows, d_0 as the category containing four objects A_0, A_1, A_2, A_3 and with arrows generated by one non-identity arrow $c' : A_2 \longrightarrow A_2$. The functor j_0 sends A to A_1, and the functor k_0 sends A_0 to I, A_1 to A, A_2 to $A \otimes A$, and A_3 to $(A \otimes A) \otimes A$, or equally, as Tc is the free strict symmetric monoidal category on c, $A \otimes (A \otimes A)$. The functor k_0 sends c' to the symmetry $A \otimes A \longrightarrow A \otimes A$.

The second diagram type, (c_1, d_1, j_1, k_1), has c_1 given by the category with three objects B_0, B_1, B_2 and non-identity arrows $j : B_0 \longrightarrow B_1$ and $m : B_2 \longrightarrow$

B_1. We shall describe the category d_1 and the functor k_1 together by giving a subcategory of Tc_1, with k_1 being understood to be the inclusion of the subcategory d_1 into Tc_1. The category d_1 has arrows given by j and m and by $m \otimes B_1$, $B_1 \otimes m$, and $j \otimes B_1$; taking domains and codomains of these arrows determine the objects of d_1

It remains to define $\phi_0 : d_0 \longrightarrow CMon$ and $\phi_1 : d_1 \longrightarrow CMon$. The former sends A_i to X_i for each i, and sends c' to s. The latter sends j to j, m to m, $m \otimes B_1$ to m_l, $B_1 \otimes m$ to m_r, and $j \otimes B_1$ to j_l.

With these definitions, $Th(CMon)$ is, up to equivalence, the free symmetric monoidal category on a commutative monoid, and equivalently, Set_f. It may also be characterised as the free category with finite coproducts on 1.

2.8 Example Let *CComon* be the sketch for a commutative comonoid. This is just the same as Example 2.7 except that the arrows are all reversed. The generic model $Th(CComon)$ is equivalent to Set_f^{op}, which may be also be characterised as the free category with finite products on 1. The construction sending a small symmetric monoidal category C to the category of commutative comonoids in C gives the cofree category with finite products on C [6]. It is equivalent to Milner's category of wirings.

2.9 Example Let *Unit* be the sketch for an object X together with a unit $j : I \longrightarrow X$. So the underlying category of *Unit* is the arrow category, $D = (0, 1)$, and the functor $\phi : 1 \longrightarrow Unit$ sends 1 to I. The generic model $Th(Unit)$ is given by the category of finite sets and injections.

2.10 Example Let *Counit* be the sketch for an object X together with a counit $c : X \longrightarrow I$. This is dual to Example 2.9. The generic model $Th(Counit)$ is equivalent to the opposite of the category of finite sets and injections.

2.11 Example Let *RBimon* be the sketch for a relational bimonoid, i.e., an object X together with both a commutative monoid structure on X and a commutative comonoid structure on X that commute with each other and for which the comultiplication followed by the multiplication gives the identity on X. The generic model $Th(RBimon)$ is then the category of finite sets and relations.

We shall outline a proof for this. Up to equivalence, the objects of $Th(RBimon)$ are given by natural numbers, as there is one generator and one must freely add symmetric monoidal structure. For maps, using commutativity of the maps in *RBimon* with respect to each other, any map can be seen to be given by a string of counits and comultiplications, followed by a string of multiplications and units. So by the (well-known) results for monoids and comonoids, each map from m to p is given by the inverse of a function from a finite set, followed by a function. To give an inverse function followed by a function amounts, up to isomorphism, i.e., renaming, to giving a matrix. The coherence axiom asserts that there can be no redundancy, in the sense that there may be at most one possible route via the inverted function and the function between any two elements, but that is exactly the condition saying that the pair forms a relation.

Later, we can prove this result by more abstract considerations, but space prevents us from including the details. The category $Th(RBimon)$ is that Plotkin proposes to use to model wiring in CCS.

2.12 Example Let *Bimon* be the same sketch as that for relational bimonoids but without the condition that the comultiplication followed by the multiplication be the identity. The generic model $Th(Bimon)$ is given by finite sets and matrices valued in the free commutative monoid N on 1.

3 The generic model as free on 1

We now have a notion of symmetric monoidal sketch S and we have a notion of the generic model $Th(S)$ of a symmetric monoidal sketch. Our leading example has S being the sketch for a commutative monoid, in which case $Th(S)$ is the category Set_f. But Set_f is also characterised by being the free category with finite coproducts on 1. The situation for this sketch is typical, so we investigate that phenomenon in this section.

3.1 Definition A *single-sorted* sketch consists of a sketch S together with an identity on objects strict symmetric monoidal functor $\iota : Th(1) \longrightarrow Th(S)$, where 1 is the sketch given by the unit category and with no diagram types.

We usually suppress the functor ι in referring to a single-sorted sketch. The single-sortedness condition trivially holds of all our leading examples. $Th(1)$ can be described explicitly; up to equivalence, it is given by the category P whose objects are natural numbers and whose maps are permutations. The condition is essentially the same as that in the formal definition of Lawvere theory with the routine generalisation from finite products to symmetric monoidal structure.

Given a single-sorted sketch S, for any small symmetric monoidal category C, composition with ι induces a forgetful functor for which we give the suggestive notation $ev_1 : Mod_s(S, C) \longrightarrow C$.

Observe that for every small symmetric monoidal category C, and for every single-sorted symmetric monoidal sketch S, the category $Mod_s(S, C)$ possesses a symmetric monoidal structure: it is not quite given pointwise. Given h and h' in $Mod_s(S, C)$, define $(h \otimes h')(1) = h1 \otimes h'1$. Now extend the definition of $h \otimes h'$ to arbitrary objects of S by induction on the complexity of the tensor product description. Finally, define $h \otimes h'$ on arrows by conjugation using the canonical isomorphisms induced by induction between $(h \otimes h')(n)$ and $h(n) \otimes h'(n)$.

It follows that $Mod_s(S, -)$ is an endofunctor on the category $SymMon_s$ of small symmetric monoidal categories and strict symmetric monoidal functors. Moreover, ev_1, i.e., composition with $\iota : Th(1) \longrightarrow Th(S)$ is a natural transformation. So $Mod_s(S, -)$ together with ev_1 form a copointed endofunctor on $SymMon_s$.

3.2 Definition Let $Mod_s(S,-)$-$Coalg$ denote the category of coalgebras for the copointed endofunctor $(Mod_s(S,-), ev_1)$. So an object of $Mod_s(S,-)$-$Coalg$ is a small symmetric monoidal category C together with a strict symmetric monoidal functor $\phi : C \longrightarrow Mod_s(S,C)$ such that $ev_1 \cdot \phi = id$, and a map in $Mod_s(S,-)$-$Coalg$ is a strict structure preserving functor.

For an example, in the case that S is the symmetric monoidal sketch $CMon$ for a commutative monoid, the category $Mod_s(S,-)$-$Coalg$ is the category of small categories with finite coproducts: an object of $Mod_s(CMon,-)$-$Coalg$ is a small symmetric monoidal category C together with, for each object x, a monoid structure on x that respects the symmetric monoidal structure of C, but that is exactly to give a diagonal and a counit, which is exactly equivalent to giving coproduct structure for reasons we shall explain in the next section.

The endofunctor $Mod_s(S,-)$ is an accessible functor, and $SymMon_s$ is a locally presentable category. So a right adjoint to the forgetful functor $Mod_s(S,-)$-$Coalg \longrightarrow SymMon_s$ necessarily exists [8]: it is given by taking a typically transfinite limit. Later, we shall study the situation more carefully. In greatest generality, the limit is complicated. But here, our primary interest is in characterising a left adjoint.

The copointed endofunctor $Mod_s(S,-)$ is very special: for that particular copointed endofunctor, the category $Mod_s(S,-)$-$Coalg$ is finitarily 2-monadic over Cat. So it follows from general category theory that the forgetful functor to Cat must have a left adjoint [9]: the main point of this section is to find conditions under which we can characterise the value of that left adjoint on 1 by $Th(S)$.

To do that, we need to add a commutativity condition on single-sorted sketches that holds of all our leading examples. When we say "need", we mean something very specific. We know how to deal with non-commutative sketches, and we intend to deal with them in further work, extending our analysis here to incorporate Frobenius and separable sketches for example. But the techniques and results are different, with the notion of coalgebra playing a less substantial role. Here we characterise $Th(S)$ in a particular way, and for this particular way, we require commutativity of S.

The commutativity condition amounts to the assertion that every arrow of the sketch commutes, in a precise sense, with every other arrow of the sketch. Recall that $Th(S)$ has the same objects as $Th(1)$, which is equivalent to P. So for natural numbers m and p, we denote by $m \times p$, the tensor product of m copies of p. So $m \times -$ is functorial in $Th(S)$.

3.3 Definition A single-sorted sketch S is *commutative* if for all maps $f : m \longrightarrow n$ and $g : p \longrightarrow q$ in S, the two maps from $m \times p$ to $q \times n$, one given by

$$m \times p \xrightarrow{m \times g} m \times q \longrightarrow q \times m \xrightarrow{q \times f} q \times n,$$

with the other dual, where the unlabelled maps are given by canonical isomorphisms in P, agree.

3.4 Proposition For any commutative single-sorted sketch S, there is a canonical strict symmetric monoidal functor $\sigma : Th(S) \longrightarrow Mod_s(S, Th(S))$ that splits ev_1, i.e., the diagram

$$\begin{array}{ccc} Th(S) & \xrightarrow{\sigma} & Mod_s(S, Th(S)) \\ & \searrow^{id} & \downarrow{ev_1} \\ & & Th(S) \end{array}$$

commutes.

Proof It follows from the definition that $\sigma(1) = \iota : S \longrightarrow Th(S)$. So $\sigma(n)$ is determined by preservation of monoidal structure. The commutativity condition is only required to prove that the evident construction of σ on a map gives a map in $Mod_s(S, Th(S))$. ■

We shall use the Proposition to characterise $Th(S)$ as the free category with specified structure on 1.

We have already observed that, using σ, one can regard $Th(S)$ as an object of $Mod_s(S, -)$-$Coalg$.

3.5 Theorem If S is a commutative single-sorted symmetric monoidal sketch, the pair $(Th(S), \sigma)$ is the free $Mod_s(S, -)$-coalgebra on 1.

Proof Let (C, ϕ) be a $Mod_s(S, -)$-coalgebra. To give a functor from 1 to C is equivalent to giving an object of C, which in turn is equivalent to giving an object $f : S \longrightarrow C$ of $Mod_s(S, C)$ such that $\phi(f1) = f$. But to give such an $f : S \longrightarrow C$ is equivalent to giving a strict symmetric monoidal functor $\bar{f} : Th(S) \longrightarrow C$ and the condition is equivalent to preservation of coalgebra structure: the forward direction is a routine verification, and the reverse is given by considering the commuting square required of a coalgebra map applied to 1. ■

This result routinely extends to 2-categorical structure. The only reason we needed the commutativity condition was in order to make $(Th(S), \sigma)$ into an object of the category $Mod_s(S, -)$-$Coalg$.

Commutativity buys us more than this as it allows considerable simplification of the description of the right adjoint to the forgetful functor $Mod_s(S, -)$-$Coalg \longrightarrow SymMon_s$.

First, the forgotten, trivial, part of the Eckmann-Hilton argument, which we analyse more thoroughly in the next section, says essentially

3.6 Lemma If S is a commutative single-sorted sketch, there is a natural transformation with C-component $\delta_C : Mod_s(S, C) \longrightarrow Mod_s(Mod_s(S, C))$ such that $Mod_s(S, -)$ together with ev_1 and δ form a comonad on the category $SymMon_s$.

3.7 Lemma For commutative single-sorted S, the pair of strict symmetric monoidal functors, $Mod_s(ev_1)$ and $(ev_1)_{Mod_s(S,C)}$, from $Mod_s(Mod_s(S,C))$ to $Mod_s(S,C)$ are jointly monomorphic in the category $SymMon_s$.

The two lemmas immediately yield

3.8 Proposition If S is a commutative single-sorted sketch, then the category of coalgebras for the copointed endofunctor $(Mod_s(S,-), ev_1)$ is equal to the category of coalgebras for the comonad $(Mod_s(S,-), ev_1, \delta)$

This implies that for commutative single-sorted sketches, the right adjoint to the forgetful functor from $Mod_s(S,-)$-$Coalg$ to $SymMon_s$ is given simply by $Mod_s(S,-)$. This is a vast simplification of the situation for an arbitrary single-sorted sketch. In the next section, we characterise the category $Mod_s(S,-)$-$Coalg$ for a class of examples including all our leading examples.

4 The Eckmann-Hilton argument

The Eckmann-Hilton [4] argument asserts that to give a group in the category of groups is equivalent to giving an Abelian group. The argument restricts to monoids, and it goes as follows.

4.1 Theorem (Eckmann-Hilton) [4] A monoid in the category Mon of monoids is exactly a commutative monoid.

Proof Let $(M, \circ, e)$ and $(M, \circ', e')$ be monoids whose operations commute with each other. Observe that $e = e'$ because the map $e : 1 \longrightarrow M$ respects the units of 1 and $(M, \circ', e')$, and the unit of 1 is the identity. Next observe that commutativity of $\circ$ with $\circ'$, together with the equality of the two units, implies that $x \circ y = y \circ' x$. Finally, use the same commutativity again but with units in different places to deduce that $x \circ y = x \circ' y$. Putting this together, we are done. ■

None of this argument is special to the situation of monoids in Set. It all holds with Set generalised to an arbitrary symmetric monoidal category. Moreover, the argument regarding units does not require even the existence of the compositions; and the argument, holding for an arbitrary symmetric monoidal category C, also holds for C^{op}, so one has the dual.

The heart of the result asserts that taking Mon to be the sketch for not necessarily commutative monoids, the category $Mod_s(Mon, Mod_s(Mon, C)$ is coherently isomorphic to the category $Mod_s(CMon, C)$, where $CMon$ is the sketch for commutative monoids that we have already introduced. We do not address the full generality of that here, as we are restricting attention to commutative sketches. In that restricted situation, we can rephrase the result as saying that δ, the comultiplication for the comonad $Mod_s(CMon, -)$, is an isomorphism. That is to say that $Mod_s(CMon, -)$ is an "idempotent" comonad.

The significance of being an idempotent comonad is that each object C of $SymMon_s$ has at most one coalgebra structure on it. That suggests that the coalgebra structure is probably given by a universal property, and indeed that is the case in our leading examples. In particular, it means that $Th(S)$ has a description in terms of a universal property.

We shall spell all this out in detail in order to capture our examples. For any small symmetric monoidal category C, write $Unit(C)$ for $Mod_s(Unit, C)$, where the sketch $Unit$ is as we defined it earlier. So an object of $Unit(C)$ consists of an object X of C together with a map $e : I \longrightarrow X$, and an arrow is a map in C that respects the units.

4.2 Proposition The forgetful functor $Unit(Unit(C)) \longrightarrow Unit(C)$ that forgets the inner structure is an isomorphism of categories.

Proof To give an object of $Unit(Unit(C))$ is to give an object $(X, j : e \longrightarrow X)$ of $Unit(C)$ together with a map $e' : I \longrightarrow X$ such that $e' \circ id = e$. Trivially, $e = e'$. ■

Here, $Mod_s(Unit, -)$-$Coalg$ is the category of small symmetric monoidal categories for which the unit is the initial object. So we may conclude that $Unit(C)$ is the cofree small symmetric monoidal category for which the unit is the initial object on C, and $Th(Unit)$ is the free such on 1.

By considering the symmetric monoidal category C^{op}, the same result holds for $Mod_s(Counit, -)$. So $Counit(C)$ is the cofree small symmetric monoidal category for which the unit is the terminal object on C, and $Th(Counit)$ is the free such on 1.

Similarly, denote $Mod_s(CMon, C)$ by $CMon(C)$ for any small symmetric monoidal category C. It is the category of commutative monoids in C.

4.3 Proposition The forgetful functor $CMon(CMon(C)) \longrightarrow CMon(C)$ forgetting the inner structure is an isomorphism of categories.

Proof This is the Eckmann-Hilton argument for a symmetric monoidal category C. ■

So $Mod_s(CMon, -)$, which agrees with $CMon(-)$, is right adjoint to the forgetful functor from the category of small categories with finite coproducts to the category of small symmetric monoidal categories, and $Th(CMon)$ is the free category with finite coproducts on 1. Taking a dual, $CComon(-)$ is right adjoint to the forgetful functor from the category of small categories with finite products to that of small symmetric monoidal categories, and $Th(CComon)$ is the free category with finite products on 1.

Putting Proposition 4.3 together with its dual, the right adjoint statements immediately extend to bimonoids and relational bimonoids. So $Bimon(-)$, with the evident definition, is right adjoint to the forgetful functor from the category of small categories with finite biproducts to the category of small symmetric monoidal categories, and $Th(Bimon)$ is the free category with finite biproducts on 1.

Similarly, $RBimon(-)$, again with the evident definition, is the right adjoint to the forgetful functor from the category of small categories with relational finite biproducts to the category of small symmetric monoidal categories, and $Th(RBimon)$ is the free category with relational finite biproducts on 1, i.e., the category of finite sets and binary relations Rel_f as studied by Plotkin.

5 Two-dimensional issues

Until now, we have restricted our attention to strict models. Here, we extend our analysis to models. The problem with strict models is illustrated by the following. One could write a symmetric monoidal sketch for a monoid identifying the base object m with $m \otimes 1$, for instance by taking a fragment of the Lawvere theory for a monoid. But in Set, a set X is not equal to $X \times 1$ in general, but only isomorphic to it. So although the models of the sketch are monoids as expected, there are no strict models of this sketch. A similar concern was central to [2].

One can avoid such a problem in an ad hoc way by simply not writing sketches like that, but being careful in writing a sketch specifically not to identify two objects that one wants to allow to be distinct in models. For instance, one could write a sketch for monoids without identifying m with $m \otimes 1$. We have done something a little more subtle than that in our definition of single-sorted sketch, as our precise definition of single-sortedness implies that m and $m \otimes 1$ must be kept separate. But traditionally, sketches such as that outlined for monoids above have been considered as reasonable, and the notion of model has been defined to allow them. So we follow suit.

We solve the problem in general by extending our definition of model to agree with the usual one in our examples, then proving that for any sketch $\mathcal{S}$, there is a sketch $\mathcal{S}'$ such that the category of models of $\mathcal{S}$ is isomorphic to the category of strict models of $\mathcal{S}'$. Thus we extend our main theorem, Theorem 2.5, from strict models to models.

For an elegant account of this, we must first extend our definition of map of $SymMonD$-sketches.

5.1 Definition A *pseudo-map* of $SymMonD$-sketches from (X, ϕ_i) to (Y, ψ_i) is a functor $f : X \longrightarrow Y$ together with, for every i, an invertible natural transformation $\bar{f}_i : \psi_i \Rightarrow f\phi_i$ such that $\bar{f}_i j_i = id$.

$SymMonD$-sketches and pseudo-maps form a category $SymMonD\text{-}Sketch_p$. In fact, both $SymMonD\text{-}Sketch$ and $SymMonD\text{-}Sketch_p$ form 2-categories. There is an evident inclusion $J : SymMonD\text{-}Sketch \longrightarrow SymMonD\text{-}Sketch_p$, and we have

5.2 Proposition The inclusion $J : SymMonD\text{-}Sketch \longrightarrow SymMonD\text{-}Sketch_p$ has a left adjoint $(-)'$.

Proof This is essentially an example of the main result of [2], and we use the notation of [2] freely. Given (X, ϕ_i), first take the pseudo-colimit $\lambda_i : p_i\phi_i \Rightarrow q_i : d \longrightarrow X''_i$ of each ϕ_i. Then take the coidentifier $r_i : X''_i \longrightarrow X'_i$ of $\lambda_i j_i$ to ensure that the equation is satisfied for each i. Finally, take the colimit $X' = \coprod_X X'_i$ of $r_i p_i$'s with injections s_i. ■

In fact, one can prove, as in [2], that every sketch $\mathcal{S}$ is equivalent, in the 2-category $SymMonD\text{-}Sketch_p$, to $\mathcal{S}'$. To define a model, let C be a small symmetric monoidal category, and let $\mathcal{S} = (X, \phi_i)$ be a $SymMonD$-sketch. For ease of notation, we shall henceforth leave the subscripts on the elements of D implicit.

5.3 Definition A *model* of (X, ϕ) in C is a functor $f : X \longrightarrow C$ together with an isomorphism

$$\begin{array}{ccc} d & \xrightarrow{\quad k \quad} & Tc \\ {\scriptstyle \phi}\big\downarrow & \Downarrow \bar{f} & \big\downarrow{\scriptstyle (f\phi j)^*} \\ X & \xrightarrow[\quad f \quad]{} & C \end{array}$$

such that $\bar{f}j = id$.

It is routine to modify the definition of $Mod_s(\mathcal{S}, C)$ to define the category $Mod(\mathcal{S}, C)$. We need to analyse that category.

5.4 Proposition Given a $SymMonD$-sketch (X, ϕ), for every model of (X, ϕ) in a small symmetric monoidal category C, there exists a unique D-structure on C such that the data for the model is exactly that for a pseudo-map of $SymMonD$-sketches, and the axioms hold.

Proof Existence is immediate from the definitions of model and pseudo-map of $SymMonD$-sketches. Unicity follows directly from the coherence condition. ■

5.5 Corollary For every $SymMonD$-sketch $\mathcal{S}$, there exists a $SymMonD$-sketch $\mathcal{S}'$ and a pseudo-map $j : \mathcal{S} \longrightarrow \mathcal{S}'$ of $SymMonD$-sketches such that composition with j yields an isomorphism of categories from $Mod_s(\mathcal{S}', C)$ to $Mod(\mathcal{S}, C)$ for every small symmetric monoidal category C.

This result solves our problem, as it may be combined with Theorem 2.5 to yield

5.6 Corollary For every $SymMonD$-sketch $\mathcal{S}$, composition with $\iota j : \mathcal{S} \longrightarrow Th(\mathcal{S}')$ yields an isomorphism of categories from $SM_s(Th(\mathcal{S}'), C)$ to $Mod(\mathcal{S}, C)$ for any small symmetric monoidal category C.

One can go a little further than this. If one wants to restrict attention to strong symmetric monoidal functors, i.e., functors that only preserve structure up to coherent isomorphism rather than strictly, then we can adopt the theory of [2] directly. If one extends $SymMon_s$ to $SymMon$, the 2-category with the same objects as $SymMon_s$, but with strong symmetric monoidal functors as 1-cells, then it follows as a consequence of the main theorem of [2] (see [10] for more detail of this) that we have

5.7 Theorem For every $SymMonD$-sketch $\mathcal{S}$, composition with $\iota j : \mathcal{S} \longrightarrow Th(\mathcal{S}')$ yields an equivalence of categories from $SM(Th(\mathcal{S}'), C)$ to $Mod(\mathcal{S}, C)$ for any small symmetric monoidal category C.

Now we have these results, it follows that $Mod(\mathcal{S}, -)$ gives a functor from $SymMon$ to Cat and hence to $SymMon$ as explained before.

References

1. M. Barr and C. Wells, Toposes, Triples, and Theories, Grundlehren der math. Wissenschaften 278, Springer-Verlag (1985).
2. R. Blackwell, G.M. Kelly, and A.J. Power, Two-dimensional monad theory. J. Pure Appl. Algebra 59 (1989) 1–41.
3. G.L. Cattani. M.P.Fiore, and G. Winskel, A theory of recursive domains with application to concurrency, Proc. LICS 98 (1998) 214–225.
4. B. Eckmann and P.J. Hilton, Group-like structures in general categories I: multiplications and comultiplications, Math. Annalen 145 (1962) 227–255.
5. M.P. Fiore, G.D. Plotkin, and A.J. Power, Cuboidal sets in axiomatic domain theory, Proc. LICS 97 (1997) 268–279.
6. T. Fox, Coalgebras and cartesian categories, Comm. Algebra (7)4 (1976) 665-667.
7. M. Hyland and A.J. Power, The structure of categories of wirings (submitted).
8. P.T. Johnstone, A.J. Power, T. Tsujishita, H. Watanabe, and J. Worrell, An axiomatics for categories of transition systems as coalgebras, Proc LICS 98 (1998) 207–213.
9. G.M. Kelly and A.J. Power, Adjunctions whose counits are coequalizers, and presentations of finitary enriched monads, J. Pure Appl. Algebra 89 (1993) 163–179.
10. Y. Kinoshita. A.J. Power, and M. Takeyama, Sketches, J. Pure Appl. Algebra 143 (1999) 275–291.
11. A. Mifsud, R. Milner, and A.J.Power, Control structures, Proc. LICS 95 (1995) 188–198.
12. A.J. Power, Elementary control structures, Proc. CONCUR 96, Lecture Notes in Computer Science 1119 (1996) 115–130.