Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER
Electronic Notes in Theoretical Computer Science 100 (2004) 31–46

Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Symmetric Monoidal Sketches and Categories of Wirings

Martin Hyland

Dept of Pure Mathematics and Mathematical Statistics
University of Cambridge
Wilberforce Road, Cambridge, ENGLAND

and

John Power[1]

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

Abstract

We introduce a potential application of two-dimensional linear algebra to concurrency. Motivated by the structure of categories of wirings, in particular in action calculi but also in other models of concurrency, we investigate the notion of symmetric monoidal sketch for providing an abstract notion of category of wirings. Every symmetric monoidal sketch generates a generic model. If the sketch is single-sorted, the generic model can be characterised as a free structure on 1, with structure defined coalgebraically. We investigate how these results generalise results about categories of wirings given by Milner and others, and we outline how the constructs may be extended to model controls and dynamics.

Keywords: category of wirings, symmetric monoidal sketch, model, generic model, coalgebra

[1] This work is supported by EPSRC grant GR/M56333: The structure of programming languages : syntax and semantics, and a British Council grant and the COE budget of STA Japan.

doi:10.1016/j.entcs.2004.09.004

1 Introduction

Over recent years, we have introduced and begun to develop an abstract theory of "two-dimensional linear algebra"[17,18,19,20]. We first explicitly used the expression, which was suggested to us by Bill Lawvere, in [18]. It amounts to the systematic replacement in linear algebra of sets by categories, involving a systematic replacement of the category Ab by a 2-category of small symmetric monoidal categories: sometimes we need maps that strictly preserve symmetric monoidal structure, giving the 2-category $SymMon_s$, and sometimes we need maps that preserve the structure only up to coherent isomorphism, giving the 2-category $SymMon$. The two-dimensional structures we have developed have begun to prove useful in denotational semantics [16], specifically in regard to the problem of giving an elegant unified account of the ways in which one might combine "notions of computation." But we believe they should also prove useful in modelling concurrency. So this paper is devoted to an attempt to outline how we think these ideas might impact on concurrency.

There are several ways to explain how two-dimensional linear algebra impacts upon concurrency. In this paper, we explain it primarily in terms of Milner's action calculi [25] because we have detailed knowledge of them. The aspect of two-dimensional linear algebra we emphasise is the definition of symmetric monoidal sketch, which we introduced and began to develop in [17]. Two-dimensional linear algebra is fundamentally part of the study of coalgebra, but we do not emphasise that here, an explanation appearing in [18]: the results of Section 4 illustrate it. The focus of this paper is concurrency, its originality lying primarily in its computing application rather than in new mathematical results.

Fundamental to the notion of action calculus as developed by Milner and colleagues [9,10,11,25,26,27] is the notion of a category of wirings. For Milner et al, that category has objects given by natural numbers, with arrows freely generated by symmetries, diagonals, and discard morphisms, all subject to coherence axioms. Composition in Milner's category models a form of data-flow, so diagonals represent the fact that data may be copied, discard morphisms represent the fact that it may be discarded, and the symmetries exist by the nature of concurrency: the monoidal structure of the category models parallel composition. Having defined a category of wirings, Milner added controls to represent specific calculi, then a rewriting relation to model reaction. We believe all three of these constructions, namely defining a wiring category, adding controls, and adding an account of dynamics, will eventually have a sound category theoretic foundation, and in the final section, we outline one proposal for it; but our primary focus here is on the structure of generalised categories of wirings.

Mathematically, Milner's category of wirings is a definitive object of study: it may be characterised, up to equivalence, as the free category with finite products on 1, also as the free strict symmetric monoidal category on a commutative comonoid, and also as the category Set_f^{op}, the opposite of the category of finite sets, with monoidal structure given by coproduct of sets. We give the definition and summarise the main results in Section 2.

Not all paradigms of concurrency have categorical composition modelling data-flow in the same sense. For instance, in modelling CCS or other calculi as Plotkin and, independently, Gardner and colleagues [9,10,11] are doing, composition works rather differently. Gardner's wiring category still models the connection between names, but composition models name fusion. In Plotkin's work, the wiring categories describe connections between input and output actions, along the lines of interaction categories [1]. So Plotkin and Gardner, for separate reasons, have not only diagonals and discards, but also codiagonals and introduction maps, as there may be different fusions for one and many wires into (as well as out of) any one port for the other. One might also consider variants in which one counts the number of wires between ports, or one allows discards but not diagonals. So we seek an account of what are the range of possible categories of wirings. Our leading example is Milner's category, but we also want to include those of Plotkin and Gardner, as well as other possibilities, for instance allowing for discards but not diagonals. The notion of a category of wirings is also implicit in the work on interaction categories [1,2], so we want to incorporate that too. And it occurs extensively elsewhere in the literature, e.g., [6,28]. So, in Section 3, we recall the definition of symmetric monoidal sketch and the basic theorem about them [17], and we explain how it allows generalisation of Milner's category of wirings to include categories of wirings introduced by other researchers.

The notion of symmetric monoidal sketch may seem familiar to readers with some knowledge of finite product sketches, but that familiarity is misleading: central to the usual account of finite product sketches is the universal property of finite products; symmetric monoidal structure does not satisfy such a universal property, so requires entirely new techniques that are not routine generalisations of those for finite products.

Every symmetric monoidal sketch S has a free strict symmetric monoidal category $Th(S)$ on it, generalising the fact that every commutative comonoid has a free strict symmetric monoidal category on it. We call this strict symmetric monoidal category together with the universal model of the sketch the *generic* model of the sketch. The generic model is characterised by the property that if C is a small strict symmetric monoidal category, the category $Mod_s(S, C)$ of models of S in 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. We list explicit descriptions of various reasonable possible sketches that have appeared in the literature, primarily in those papers and from those sources cited above, and we characterise the various constructions in familiar terms in particular cases. Our analysis includes examples such as commutative monoids, commutative comonoids and relational bimonoids, and may also relate to Winskel's work using path categories [5]. It has surprised us how much there is in common here beyond the vague idea that this is all given by symmetric monoidal structure together with some added data and axioms. We remark in passing that other authors, notably Grandis and colleagues [12,13,14], have also focused on a notion in the spirit of generic model in modelling concurrency, albeit in a somewhat different setting to ours.

In Section 4, we show that, under a single-sortedness condition, $Th(S)$ can be characterised as a free coalgebra on 1, generalising the fact that Milner's construction is that of the free category with finite products on 1: the central fact here is that every object of $Th(S)$ has a canonically given coalgebra structure on it, so for instance, in Milner's category of wirings, every object has a canonically given commutative comonoid structure on it. This section contains originality in the mathematics in addition to its computational content: the results here extend those of [17] and [18].

Controls and dynamics are at the heart of modelling interaction, but one cannot model interaction until one has modelled wiring, and to date, we are unaware of any general category theoretic approach to modelling the various choices for wiring. To support our treatment, in Section 5, we show one way that one might add controls and dynamics generalising Milner's ideas to allow other concurrency paradigms.

For most of this paper, for ease of exposition, we shall gloss over coherence questions relating to the distinction between preservation and strict preservation of category theoretic structure: every monoidal category is equivalent to a strict monoidal category, so we may safely conflate the two notions. Ultimately, the relevant coherence issues may be resolved by reference to [3] and [22].

2 Categories of wirings

In this section, we investigate categories of wirings and their structure. Our leading example arises from the work on action calculi and its semantic models (see [10,11,25,26,27] and the papers cited therein). A fundamental part of the definition of an action calculus was that of its underlying wiring category. Assuming, as was the case in the leading examples, that the set of primes was given by a singleton, the wiring category was defined as follows:

Example 2.1 The *wiring category* of action calculus is the category with set of objects N, with arrows generated by symmetries $n \otimes m \cong m \otimes n$, by diagonals $\delta : n \longrightarrow n \otimes n$, and by discards $! : n \longrightarrow 0$, all subject to coherence equations, where $\otimes$ is the sum of natural numbers.

The idea is as follows. In action calculus, composition in the category of actions is used to model data-flow, and tensor product is used to model parallel composition. Given a piece of data, one can copy it or one can discard it. Parallel composition is, by its nature, symmetric. Thus one has symmetries, diagonals, and discards. The following fact was recognised, although perhaps not explicitly stated, in the action calculus work.

Proposition 2.2 *The wiring category of action calculus is the free category with finite products on the unit category* 1.

It is in the category theoretic folklore that one can further give an explicit description of the wiring category, up to equivalence, as follows.

Proposition 2.3 *The wiring category of action calculus is equivalent to the category* Set_f^{op}.

We can give a further characterisation of the category of wirings, involving some more sophisticated category theory as follows. In the definition of strict monoidal category, one has a monoid $Ob(C)$ of objects of the category. But one also has a notion of monoid in a monoidal category. We shall restrict attention to commutative monoids in a symmetric strict monoidal category.

Definition 2.4 Given a symmetric strict monoidal category C, a *commutative monoid* in C consists of an object X of C together with maps $j : I \longrightarrow X$ and $m : X \otimes X \longrightarrow X$, subject to commutativity of the evident diagrams.

Proposition 2.5 *The category of wirings of action calculus is the initial strict symmetric monoidal category with a commutative comonoid in it, i.e., for any strict symmetric monoidal category C with a commutative comonoid $(X, \delta, !)$ in C, there is a unique strict symmetric monoidal functor from the category of wirings to C that preserves the commutative comonoid structure on* 1.

These characterisations provide mathematical evidence of the definitiveness of the notion of wiring category for the action calculus. But suppose one wants to vary the notion of wiring, for instance, using composition in the category to model the physical linkage of wires rather than data-flow, such as one may use in modelling CCS. Then one might have codiagonals $\nu : n \otimes n \longrightarrow n$

and $introduce : 0 \longrightarrow n$, as in current work of Gordon Plotkin, Philippa Gardner, and others [10,11]. One may make various choices about what axioms to assert too: for instance, does one want to count the number of wires between ports, or simply the connectivity? So we seek a general account of the possibilities for categories of wiring, such that generalisations of the above results hold. That provides mathematical justification for favouring some choices of categories of wirings over others, and for asking questions such as whether one wants a trace in the category of wirings. (In some categories of wirings, though not in the original form of action calculi, a trace could be used to model iteration.)

The key notion we require is that of symmetric monoidal sketch and model of such, generalising the notion of commutative comonoid in a strict symmetric monoidal category. The central result we investigate is the coincidence between the initial strict symmetric monoidal category with a commutative comonoid in it with the free category with finite products on 1.

3 The definition of a symmetric monoidal sketch

In this section, we define the notion of a symmetric monoidal sketch, and we define its category of models in any strict symmetric monoidal category. The leading example of such a category of models is the category of commutative comonoids in a strict symmetric monoidal category as in Section 2.

Definition 3.1 A *family D of diagram types* is a small family of 4-tuples of the form $(c_i, d_i, j_i\colon c_i \to d_i, k_i\colon d_i \to Tc_i)$, where c_i and d_i are finitely presentable categories, Tc_i is the free strict 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:

$$\begin{tikzcd}
d \arrow[rr, "k"] & & Tc \\
& c \arrow[ul, "j"] \arrow[ur, "\eta_c"'] &
\end{tikzcd}$$

where η is the unit of T.

Definition 3.2 A *symmetric monoidal sketch S* consists of a small category X together with a family D of diagram types and a D-indexed family of functors $\phi_i : d_i \longrightarrow X$.

We typically denote the underlying category X of a sketch S by S itself, and we usually denote a sketch by (S, ϕ_i), not explicitly mentioning D in the notation.

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

Definition 3.3 A *strict model* of (S, ϕ) in C is a functor $f : S \longrightarrow C$ such that the following diagram commutes:

$$\begin{tikzcd}
d \arrow[r, "k"] \arrow[d, "\phi"'] & Tc \arrow[d, "(f\phi j)^*"] \\
S \arrow[r, "f"'] & C
\end{tikzcd}$$

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

For this paper, we shall simply refer to these as models rather than strict models. One can routinely define the notion of a map of models between two models of S in a strict symmetric monoidal category C: a map of models is a natural transformation that respects the structure of S. Models and maps of models yield a category $Mod_s(S, C)$. For small strict symmetric monoidal categories B and C, there is a homcategory $SM_s(B, C)$ as usual. The central result of [23] yields

Theorem 3.4 *For any symmetric monoidal sketch S, there is a small strict 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.

Example 3.5 Let $CMon$ be the sketch for a commutative monoid. Details appear in [17]. The category $CMon$ has four objects X_0, X_1, X_2, and X_3, with arrows 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 evident diagrams.

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 rest of the data is similar or evident.

$Th(CMon)$ is the free strict symmetric monoidal category on a commutative monoid, which is equivalent to Set_f with symmetric monoidal structure given by finite coproduct of sets. It may also be characterised as the free category with finite coproducts on 1.

Example 3.6 For the sketch $CComon$ for a commutative comonoid, take the dual of the sketch for a commutative monoid. 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. This is exactly Milner's category of wirings [25,26,27]. Also observe that the construction sending a small strict symmetric monoidal category C to the category of commutative comonoids in C gives the cofree category with finite products on C [8].

Example 3.7 Let $Zero$ be the sketch for an object X with a unit $j : I \longrightarrow X$ and a counit $c : X \longrightarrow I$ which commute with each other in the sense that $c.j = id_I$. The generic model $Th(Zero)$ is equivalent to the category of finite sets and partial bijections. This category occurs in the Geometry of Interaction (see [1,2] and the references therein).

Example 3.8 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, that is, $m.\delta = id_X$. The generic model $Th(RBimon)$ is then the category of finite sets and relations. For a proof of this, see [17]. The category $Th(RBimon)$ is the one Plotkin proposes to use to model wiring in CCS. It is also being studied by Gardner [10].

Example 3.9 Let $RFrob$ be the sketch for relational Frobenius objects, i.e., the same sketch as $RBimon$ except that the monoid and comonoid structures need not commute with each other, but rather one has commutativity of the diagram

$$\begin{tikzcd}
X \otimes X \arrow[r, "X \otimes \delta"] \arrow[d, "m"'] & X \otimes (X \otimes X) \arrow[d, "m \otimes X"] \\
X \arrow[r, "\delta"'] & X \otimes X
\end{tikzcd}$$

and $m.\delta = id_X$. The generic model $Th(RFrob)$ is given by finite sets, and with a map from m to n given by an equivalence relation on $m+n$. This category is implicitly used by Danos and Regnier [7] in connection with the Geometry of Interaction and is considered by Gardner in [10] for quite different reasons.

There are several variants of this example. In particular, dropping the condition $m.\delta = id_X$ yields Frobenius objects, whose generic model is characterised in [4]. Further, in [10], Gardner suggested considering diagonals, discards, and introduction morphisms, but with no codiagonals as this gives an exact account of the π-calculus as an action calculus in the sense that the structural congruences match.

4 The generic model as a free coalgebra 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 comonoid, in which case $Th(S)$ is the category Set_f^{op}. And Set_f^{op} is also characterised as the free category with finite products on 1. We generalise these phenomena in this section.

Definition 4.1 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 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 isomorphism, it is given by the category P whose objects are natural numbers and whose maps are permutations. The single-sortedness 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 strict symmetric monoidal category C, composition with ι induces a forgetful functor

$$Mod_s(S,C) \cong SM_s(Th(S),C) \longrightarrow SM_s(Th(1),C) \cong Mod_ss(1,C) \cong C$$

for which we give the suggestive notation $ev_1 : Mod_s(S,C) \longrightarrow C$.

Observe that for every small strict symmetric monoidal category C, and for every single-sorted symmetric monoidal sketch S, the category $Mod_s(S,C)$ possesses a strict 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 $h(n) \otimes h'(n)$.

So far, we have focused on the freeness of our constructions, in particular in considering the generic model. But freeness is quite a common phenomenon when dealing with categories with given structure. Typically much more profound, and a phenomenon that is distinctive of linear algebera as opposed to universal algebra in general, is the situation where one has not only freeness but also cofreeness, i.e., one has coalgebraic structure. So we explore that now.

Definition 4.2 A *copointed endofunctor* on a category D consists of a functor $H : D \longrightarrow D$ and a natural transformation $\epsilon : H \Rightarrow Id$. An (H,ϵ)-*coalgebra* consists of an object X of D together with a map $\phi : X \longrightarrow HX$ such that $\epsilon_X \cdot \phi = id_X$. A *map of coalgebras* is a map in D that respects the coalgebra structure.

Coalgebras and maps of coalgebras form a category we denote by (H,ϵ)-$Coalg$. We typically drop ϵ from the notation if it is clear.

The construction $Mod_s(S,-)$ extends to an endofunctor on the category $SymMon_s$ of small strict 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 $SymMon_s$.

Example 4.3 If S is the symmetric monoidal sketch $CComon$ for a commutative comonoid, the category $Mod_s(S,-)$-$Coalg$ may be characterised, up to the natural 2-categorical notion of equivalence, as the category of small categories with finite products and functors that strictly preserve finite products: an object of $Mod_s(CComon,-)$-$Coalg$ is a small strict symmetric monoidal category C together with, for each object x, a commutative comonoid structure on x that respects the symmetric monoidal structure of C, but that is exactly to give a codiagonal and a discard, which is exactly equivalent to giving product structure for reasons we shall explain later.

We still need one final refinement of this construction in order to prove our main theorem. The problem is that the category $Mod_s(S,C)$ has too few maps, and that stops us from making $Th(S)$ into an object of $Mod_s(S,-)$-$Coalg$: although we can readily build a function from the set of objects of $Th(S)$ to the set of objects of $Mod_s(S,Th(S))$, and that function respects the monoidal structure, we cannot in general extend that function to become a functor (although we can do so in our leading example). The most general solution to this is to modify the definition of $Mod_s(S,-)$-$Coalg$.

Definition 4.4 Given a single-sorted sketch S and a strict symmetric monoidal category C, let $Mod_s^*(S,C)$ denote the (unique) factorisation

$$Mod_s(S,C) \longrightarrow Mod_s^*(S,C) \longrightarrow C$$

of $ev_1 : Mod_s(S,C) \longrightarrow C$ into a functor $Mod_s(S,C) \longrightarrow Mod_s^*(S,C)$ that is the identity on objects followed by a fully faithful functor $Mod_s^*(S,C) \longrightarrow C$.

This construction is more definitive than it may appear, and has already proved useful in modelling the combining of notions of computation [16], as we have the following result. We leave implicit the definition of a category of single-sorted sketches.

Proposition 4.5 *Given single-sorted sketches S and S', the sum $S+S'$ is characterised by*

$$Mod_s^*(S+S',C) \cong Mod_s^*(S,Mod_s^*(S',C))$$

natural in C.

The construction $Mod_s^*(S,C)$ extends to an endofunctor on $SymMon_s$ and ev_1 trivially restricts to provide a copoint ev_1^* for the endofunctor. Thus we have the category $Mod_s^*(S,-)$-$Coalg$ of coalgebras for the copointed endofunctor $(Mod_s^*(S,-),ev_1^*)$. A right adjoint to the forgetful functor $Mod_s^*(S,-)$-$Coalg \longrightarrow SymMon_s$ exists for general reasons [21]. 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, it follows from general category theory that the forgetful functor to Cat must have a left adjoint [22]: the main point of this section is to find conditions under which we can characterise the value of that left adjoint on 1 by $Th(S)$.

Proposition 4.6 *For any single-sorted symmetric monoidal 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*

$$\begin{tikzcd}
Th(S) \arrow[r, "\sigma"] \arrow[dr, "id"'] & Mod_s^*(S, Th(S)) \arrow[d, "ev_1^*"] \\
& Th(S)
\end{tikzcd}$$

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 behaviour of σ on maps is necessarily the identity. □

Theorem 4.7 *If S is a single-sorted symmetric monoidal sketch, then the pair $(Th(S), \sigma)$ is the free $Mod_s^*(S, -)$-coalgebra on 1.*

Proof. By Proposition 4.6, using σ, one can regard $Th(S)$ as an object of the category $Mod_s^*(S, -)$-$Coalg$. 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 an $f : S \longrightarrow C$ is equivalent to giving a strict symmetric monoidal functor $\bar{f} : Th(S) \longrightarrow C$ and the condition $\phi(f1) = f$ 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. □

That constitutes the main result of this section and completes our analysis of categories of wiring. In the final section of the paper, we give a tentative account of how to use our category theoretic formulation of a category of wirings as a basis on which to give a unified account of several approaches to the modelling of concurrency, with Milner's and Plotkin's approaches as leading examples for us.

5 Controls and dynamics

In the earlier sections of the paper, we have given a category theoretic foundation for wiring. But concurrency involves far more than wiring. Specifically, we need to add accounts of controls and dynamics. There are several choices here, and it is not entirely clear to us which will prove to be definitive. But we do have several well-studied examples to guide us, specifically in [24], [27], and [15].

In [27], it was shown that to give the underlying category of one of Milner's control structures is equivalent, modulo two caveats, to giving the underlying category of an elementary control structure, which is defined as follows.

Definition 5.1 The underlying category of an *elementary control structure* consists of

- Milner's wiring category W
- a strict symmetric monoidal category C together with an identity on objects strict symmetric monoidal functor $J : W \longrightarrow C$
- for each control K, a function of the form

  $$C(k \otimes m_1, n_1) \times \cdots \times C(k \otimes m_r, n_r) \longrightarrow C(k \otimes m, n)$$

  natural with respect to maps $f : k \longrightarrow k'$ in W.

In Hasegawa's Distinguished Dissertation [15] (see also [24]), it was shown how to use this category theoretic formulation of Milner's control structures to incorporate an account of recursion by use of Milner's controls and axioms for defining the notion of *reflexive* control structure. This is given as follows.

Example 5.2 For each arity k, have a control which, to each action from $k \otimes m$ to $k \otimes n$, yields an action from m to n. Subject these to axioms for naturality and coherence in k, m and n, with respect to both categorical composition and the tensor product. Controls for a control structure are equally controls for an elementary control structure. So the equivalent version of Mifsud's reflexive control structures in terms of elementary control structures amounts to elementary control structures with what is called a *trace* on the strict symmetric monoidal category C. The concepts of reflexion and trace are studied in detail in terms of control structures in Mifsud's thesis [24] and in terms of elementary control structures in [15].

The significance of this example for us is that it indicates what structure to seek in order to model controls, as we have the following category-theoretic results.

Theorem 5.3 *The category of small traced symmetric monoidal categories is monadic over $SymMon_s$.*

Theorem 5.4 *The category of small traced symmetric monoidal categories is not monadic over Cat.*

A proof of the former theorem follows readily from the work of [22], but there appears to be no detailed account in the literature at present. One can prove the latter result by following the definitions carefully and by characterising the category of algebras generated by the adjunction

$$F \dashv U : Trace \longrightarrow Cat$$

induced by the forgetful functor: an algebra for the monad is given by a small symmetric monoidal category C together with, for each arrow $f : X \longrightarrow X$ in C, a *trace* $Tr_x : I \longrightarrow I$, subject to coherence conditions. Again, there does not seem to be a detailed analysis of this in the literature either.

It is routine, using [22], to verify that every class of Milner's controls, together with equational axioms, yields a monad T_K on $SymMon_s$. So, given a category of wirings W, a tentative general notion of control is given by a monad T_K on $SymMon_s$ and a tentative notion of a model of the control T_K is given as follows.

Definition 5.5 A T_K-model consists of

- a wiring category W
- a T_K-algebra C together with an identity on objects strict symmetric monoidal functor $J : W \longrightarrow C$
- the T_K-algebra action on C.

Example 5.6 The initial control structure (= action calculus) for a version of the π-calculus without replication has actions generated by controls

$$\boldsymbol{\nu} : 1 \to p \qquad \mathbf{out} : p \otimes m \to 1 \qquad \frac{a : m \to n}{\mathbf{box}(a) : p \to n}$$

These yield a new name of arity p, output through a port, and input modelled by means of $\mathbf{box}(id_m)$. So one has the elementary control structure freely generated by these controls. The actions of this elementary control structure consist of the closed terms for a fragment of the π-calculus, together with some closed formulae that have no computational meaning but which are generated by the syntax of the π-calculus. For more detail of this and other variants of the π-calculus as action calculi and hence as elementary control structures, see [24].

Finally, for dynamics, one has at least two main choices. Milner [25,26] modelled dynamics by means of a preorder to model reaction, which in turn was adapted for elementary control structures [27] and which can routinely be added here simply by insisting that every category in sight be enriched with a coherent partial order on each homset, respecting all the structure in sight except that for the controls (one specifically wants controls to be able to kill reaction: consider the λ-calculus or the π-calculus, both of which do not allow reaction under some binders), subject to naturality conditions. Plotkin, in contrast, is opting for an operational semantics, which has quite different structure. That too can be modelled here, but in a fundamentally different way.

let Cat_O denote the category of small categories and identity on objects functors. Given a category of wirings, one must consider the slice category W/Cat_O, for which an object consists of a small category C together with an identity on objects functor $J : W \longrightarrow C$. This category has finite products, which we characterise below. So we may consider a monoid M in W/Cat_O, and consider an M-action on a given control category C. We have not developed the general analysis in sufficient detail to bear reporting here, but to give an indication of what we have in mind, consider the following proposition.

Proposition 5.7 *For a fixed wiring category W, the category W/Cat_O has finite products given as follows: the product $J_M \times J_C$ of $J_M : W \longrightarrow M$ with $J_C : W \longrightarrow C$ has objects given by the objects of W, and with homset $(J_M \times J_C)(w, w')$ given by $M(w, w') \times C(w, w')$; the terminal object is evident.*

This means that an action of J_M on J_C is exactly as one expects, respecting the structure of the wiring. Such an analysis accounts for CCS as Plotkin wants, but, a priori, it is too restrictive to account for the π-calclulus, as the latter allows reaction to alter wiring. We can, of course, extend this formalism to account for such alteration in wiring, but the point here is that this formalism allows us to make a precise distinction between those calculi whose dynamics admit alteration of wiring and those calculi that do not. There are easy theorems (see for example [3]) that show that this structure is coherent relative to symmetric monoidal structure.

References

[1] Abramsky, S., *Interaction categories*, "Proc. Theory and Formal Methods **93**," Workshops in Computer Science, Springer-Verlag (1993) 57–70.

[2] Abramsky, S., *Retracing some paths in process algebra*, "Proc. CONCUR **96**," Lecture Notes in Computer Science **1119** (1996) 1–17.

[3] Blackwell, R., G.M. Kelly, and A.J. Power, *Two-dimensional monad theory*, J. Pure Appl. Algebra **59** (1989) 1–41.

[4] Carmody, S.M., "Cobordism Categories," Ph D dissertation, Cambridge (1996).

[5] Cattani, G.L., M.P.Fiore and G. Winskel, *A theory of recursive domains with application to concurrency*, "Proc. LICS **98**," IEEE Press (1998) 214–225.

[6] Corradini, A., and F. Gadducci, *An Algebraic Presentation of Term Graphs via GS-Monoidal Categories*, Applied Categorical Structures **7** (1999).

[7] Danos, V., and L. Regnier, *The Structure of Multiplicatives*, Arch. Math. Logic **28** (1989) 181–207.

[8] Fox, T., *Coalgebras and cartesian categories*, Comm. Algebra **(7)4** (1976) 665–667.

[9] Gardner, P., *Closed action calculi*, Theoretical Computer Science **228** (1999) 77–103.

[10] Gardner, P., *Graphical presentations of interactive systems*, MathFit Summer School (1998).

[11] P. Gardner and L. Wischik, *Explicit Fusions*, "Proc. MFCS **2000**," Lecture Notes in Computer Science **1893** (2000) 373–382

[12] Grandis, M., *Ordinary and directed combinatorial homotopy, applied to image analysis and concurrency*, Homology Homotopy Appl. **5** (2003) 211–231.

[13] Grandis, M., *Finite sets and symmetric simplicial sets*, Theory Appl. Categ. **8** (2001) 244–252.

[14] Grandis M., and L. Mauri, *Cubical sets and their site*, Theory Appl. Categ. **11** (2003) 185–211.

[15] Hasegawa, M., "Models of sharing graphs (A categorical semantics of Let and Letrec)," Distinguished Dissertations in Computer Science, Springer-Verlag (1999).

[16] Hyland, M., G.D. Plotkin, and A.J. Power, *Combining computational effects: commutativity and sum*, "Proc. Theoretical Computer Science **2002**" (2002) 474–484.

[17] Hyland, M., and A.J. Power, *Symmetric monoidal sketches*, "Proc. PPDP **00**," ACM Press (2000) 280–288.

[18] Hyland, M., and A.J. Power, *Two-dimensional linear algebra*, "Proc. CMCS **01**," Electronic Notes in Theoretical Computer Science **44** (2001).

[19] Hyland, M., and A.J. Power, *Pseudo-commutative monads*, "Proc. MFPS **01**," Electronic Notes in Theoretical Computer Science **45** (2001).

[20] Hyland, M., and A.J. Power, *Pseudo-commutative Monads and Pseudo-closed 2-categories*, J. Pure Appl. Algebra **175** (2002) 141–185.

[21] Johnstone, P.T., A.J. Power, T. Tsujishita, H. Watanabe, and J. Worrell, *An axiomatics for categories of transition systems as coalgebras*, "Proc LICS **98**," IEEE Press (1998) 207–213.

[22] Kelly, G.M., and A.J. Power, *Adjunctions whose counits are coequalizers, and presentations of finitary enriched monads*, J. Pure Appl. Algebra **89** (1993) 163–179.

[23] Kinoshita, Y., A.J. Power, and M. Takeyama, *Sketches*, J. Pure Appl. Algebra **143** (1999) 275–291.

[24] Mifsud, A., "Reflexive control structures," Edinburgh Ph.D. thesis.

[25] Milner, R., *Calculi for interaction*, Acta Informatica **33** (1996) 707–737.

[26] Mifsud, A., R. Milner, and A.J. Power, *Control structures*, "Proc. LICS **95**," IEEE Press (1995) 188–198.

[27] Power, A.J., *Elementary control structures*, "Proc. CONCUR **96**," Lecture Notes in Computer Science **1119** (1996) 115–130.

[28] Stefanescu, G., "Algebra of Flownomials," Technical University of Munchen, Institut fur Informatik (1994).