On the algebra of feedback and systems with boundary

Piergiulio Katis[1], N. Sabadini[2], R.F.C.Walters[1]

[1] School of Mathematics and Statistics, University of Sydney, Australia
[2] Dipartimento di Scienze dell'Informazione, Universitá di Milano, Italy

July, 1999

Abstract. The notion of a category-with-feedback is introduced and two main classes of examples, constructed from input-feedback-output systems and spans of graphs, are described in detail. After surveying a range of applications to computer science, we concentrate on two new examples of the algebra: continuous linear systems, which includes a model of RLC circuits; and a formalization of double-entry accounting and other systems satisfying the continuity equation.

1 Introduction

The authors have introduced elsewhere [24], [25] two complementary algebras for systems with boundaries: the first is the bicategory of circuits (or input-feedback-output systems) in the monoidal category $(\mathsf{Set}, \times)$ in which the natural feedback operation has delay; and the second is the compact closed bicategory of spans of graphs in which the natural feedback is instantaneous. The former provides a model of deterministic input-driven systems (such as asynchronous circuits), while the latter provides a model of non-deterministic systems (and also specifications of systems). These two bicategories are representatives of two broader classes of algebras and, generally, there are homomorphisms from the first type of algebra to the second.

Both these classes of algebras may be studied in the context of *bicategories-with-feedback*, a notion we will not define in this paper. We will, however, consider specific bicategories-with-feedback and define the notion of a *category-with-feedback*, which includes traced symmetric monoidal categories ([22]). In this context, we will consider functors between categories-with-feedback – functors which model and facilitate the interplay between systems and their behaviours which occurs when specifying, building and analysing machines or programs.

To each of these algebras there is an associated geometry. That is, to an expression in the algebra there is an associated geometric presentation, which corresponds to the distributed nature of the system. (When considering categories of circuits, the usual circuit diagrams are obtained; in the context of algorithms, flow charts with feedback are obtained.) This relation between algebra and geometry is in the line of the Penrose diagrams [39] for the tensor calculus, and the more recent developments in geometry and physics on braided tensor categories [21], Frobenius algebras, cobordism etc. This connection between algebra and geometry is being exploited to provide a graphical user interface for a computer program [17] written by Robbie Gates for calculating in **Span**(**Graph**). For our present purposes, we will only indicate how pictures of expressions can be drawn, deferring a formal treatment of this geometric presentation to another paper.

Over the last few years, compact closed and traced monoidal categories have received some attention from theoretical computer scientists, a notable example being Samson Abramsky's work on Interaction Categories and Game Semantics ([1], [2]). In most of these models, the internal state and dynamics of systems are abstracted away and the interfaces are endowed with more structure in order to model particular paradigms of communication. In contrast to this, the theory here described permits an explicit description of the internal dynamics of systems as well as the process of abstraction needed in order to specify deterministic systems. For example, the bicategory of ifo systems in $(\mathbf{Set}, \otimes)$ describes deterministic systems whose boundaries only carry enough structure to describe 'flows' or transitions of data. By passing to the algebra **Span**(**Graph**), however, one can describe more abstract specifications for such systems (such as communication protocols) – and, moreover, what it means for systems to satisfy such specifications.

It is the purpose of this note to give a brief introduction to and demonstrate the wide range of applicability of these algebras. After surveying the range of examples we concentrate on two new ones: in Section 4, we show that continuous linear systems are an example of the algebra, which includes a model of RLC circuits, and we indicate how the model can be extended to incorporate more general continuous systems; and in Section 5, a formalization of double-entry accounting and other systems satisfying the continuity equation is given.

We acknowledge the support for this project by the Australian Research Council, Italian MURST 40% and the Italian CNR. This paper crystallized during the Summer School on Category Theory and Computer Science held at Mount Alison University (New Brunswick, Canada) in June 1998, and we would like to thank the organizers and participants of this event. We also want to thank Bill Lawvere and the referee for suggestions which have improved the clarity of this paper.

2 Input-feedback-output systems

It has long been recognised that compact closed categories admit a feedback operation or a trace ([31]). That braided compact categories can be equipped with feedback operations has also been recognised in the context of knot theory. The first axiomatic definition of a category equipped with a feedback operation was that of an iteration theory ([5]). More recently, motivated by the concerns of the geometry of tensor calculus, the notion of a traced monoidal category ([22]) has been introduced.

Over the last few years, the authors (motivated mostly by applications to Computer Science and Engineering) have developed the notion of a bicategory-with-feedback which is closely related to traced monoidal categories, though there is more structure in that bicategories-with-feedback admit a notion of delay. We begin this section by defining an important class of bicategories-with-feedback, namely the bicategories of ifo systems.

2.1 The algebra of ifo systems

Let $(\mathbf{C}, \otimes)$ be a monoidal category. The bicategory $\mathsf{Ifo}(\mathbf{C}, \otimes)$ of *ifo systems in* $(\mathbf{C}, \otimes)$ has the same objects as $\mathbf{C}$; and an arrow from X to Y in $\mathsf{Ifo}(\mathbf{C}, \otimes)$ is a pair (U, α) where $\alpha : X \otimes U \to U \otimes Y$ is a map in $\mathbf{C}$. A 2-cell from $f : (U, \alpha) \Rightarrow (V, \beta) : X \to Y$ is an arrow $f : U \to V$ of $\mathbf{C}$ such that the following diagram commutes.

$$\begin{tikzcd} XU \arrow[r, "\alpha"] \arrow[d, "Xf"'] & UY \arrow[d, "fY"] \\ XV \arrow[r, "\beta"'] & VY. \end{tikzcd}$$

In the above, juxtaposition denotes the tensor product. We will adopt this convention elsewhere in this paper. The identity arrow at X is (I, θ) where I is the unit for $\otimes$ and $\theta : XI \to IX$ is the obvious isomorphism. The composite of $(U, \alpha) : X \to Y$ and $(V, \beta) : Y \to Z$ is the arrow $(UV, (U\beta) \cdot (\alpha V)) : X \to Z$. Horizontal and vertical composition of 2-cells are defined in the obvious way.

We call an arrow of $\mathsf{Ifo}(\mathbf{C}, \otimes)$ an *input-feedback-output system* in $(\mathbf{C}, \otimes)$, or an ifo system for short. (In [24], such a structure was called a *process* in $(\mathbf{C}, \otimes)$.) Given an ifo system $(U, \alpha) : X \to Y$, we refer to X as the *input*, U as the *internal state* and Y as the *output* of the system. The morphism α is called the *dynamics* of the ifo system.

A discrete-time, deterministic dynamical system in a category $\mathbf{C}$ is usually defined to be an endomorphism in $\mathbf{C}$ ([36]). Notice that an ifo system with input and output I is essentially such a dynamical system. Later in the paper, we will see how an ifo system in $(\mathsf{Set}, \times)$ can be interpreted as a discrete-time, deterministic input-driven system.

In [24], we called $\mathsf{Ifo}(\mathbf{C}, \otimes)$ the bicategory of loops in the suspension of $(\mathbf{C}, \otimes)$ since it can be decomposed into the composite two constructions Σ and Ω (described in the following paragraph) which are reminiscent of the suspension and loop constructions in algebraic topology.

Given a monoidal category $(\mathbf{C}, \otimes)$, define its *suspension* $\Sigma(\mathbf{C}, \otimes)$ to be the one object bicategory whose only hom-category is $\mathbf{C}$, with horizontal composition given by the tensor $\otimes$. Define the bicategory $\Sigma\mathcal{B}$ of *loops* in a bicategory $\mathcal{B}$ to comprise the bicategory of homomorphisms, oplax transformations and modifications from the additive monoid of natural numbers $\mathbf{N}$ to $\mathcal{B}$. A homomorphism from $\mathbf{N}$ to $\mathcal{B}$ is essentially a loop (that is, an endomorphism) in $\mathcal{B}$ in the following sense. Given an endomorphism $e : X \to X$ of $\mathcal{B}$, there is a homomorphism $\overline{e} : \mathbf{N} \to \mathcal{B}$ with $\overline{e}(0) = 1_X$ and, for all $n \in \mathbf{N}$, $\overline{e}(n+1) = \overline{e}(n) \cdot e$. For every homomorphism $H : \mathbf{N} \to \mathcal{B}$ there is an equivalence $H \simeq \overline{H(1)}$ in $\Sigma\mathcal{B}$. It is clear that there is a biequivalence $\mathsf{Ifo}(\mathbf{C}, \otimes) \sim \Omega(\Sigma(\mathbf{C}, \otimes))$.

As an aside, we mention that it is possible to replace $\mathbf{N}$ in the above suspension-loop construction by other categories and still obtain sensible interpretations of the arrows of the resulting bicategories as dynamical systems with input and output. For example, using the additive monoid of non-negative reals rather than the additive monoid of natural numbers yields a bicategory whose arrows can be interpreted as systems governed by continuous time rather than discrete time. The relation between these ideas and the way spans of categories and duration functors were used in [33] can be found in [23].

If $(\mathbf{C}, \otimes)$ admits a symmetry $s_{X,Y} : XY \to YX$ then the category $\mathsf{Ifo}(\mathbf{C}, \otimes)$ can be equipped with a symmetric monoidal structure: the tensor product of $(U, \alpha) : W \to Y$ and $(V, \beta) : X \to Z$ is

$$(UV, (Us_{Y,V}Z) \cdot (\alpha \otimes \beta) \cdot (Ws_{X,U}V)) : WX \to YZ.$$

In fact, the construction of ifo systems defines a functor

$$\mathsf{Ifo} : \mathsf{SymMonCat} \to \mathsf{SymMonBicat},$$

where $\mathsf{SymMonCat}$ denotes the category of symmetric monoidal categories and symmetric strong monoidal functors and $\mathsf{SymMonBicat}$ denotes the category of symmetric monoidal bicategories and symmetric strong monoidal homomorphisms.

We can also equip $\mathsf{Ifo}(\mathbf{C}, \otimes)$ with a *feedback* operation: given an arrow $(U, \alpha) : XY \to YZ$, define $\mathsf{Fb}^Y_{X,Z}(U, \alpha) : X \to Z$ to be $(YU, (s_{U,Y}Z) \cdot \alpha)$. Given a 2-cell $f : (U, \alpha) \Rightarrow (V, \beta) : XY \to YZ$, define the 2-cell $\mathsf{Fb}^Y_{X,Z} : \mathsf{Fb}^Y_{X,Z}(U, \alpha) \Rightarrow \mathsf{Fb}^Y_{X,Z}(V, \beta)$ to be Yf. It is straightforward to check that this defines a functor

$$\mathsf{Fb}^Y_{X,Z} : \mathsf{Ifo}(\mathbf{C}, \otimes)(XY, YZ) \to \mathsf{Ifo}(\mathbf{C}, \otimes)(X, Z).$$

Note that this is different to the feedback operation defined in [24].

For any $(\mathbf{C}, \otimes)$ there is a homomorphism

$$\mathsf{Inst} : (\mathbf{C}, \otimes) \to \mathsf{Ifo}(\mathbf{C}, \otimes)$$

which is the identity function on objects and which maps an arrow $f : X \to Y$ to the ifo system $(I, i \cdot f \cdot j) : X \to Y$, where $j : X \otimes I \to X$ and $i : Y \to I \otimes Y$ are the canonical isomorphisms. Note that Inst is injective on arrows. We call an arrow of the form $\mathsf{Inst}(f)$ an *instantaneous* ifo system.

When feeding back an output U as in $\mathsf{Fb}^U_{X,Y}(V,\alpha) : X \to Y$, we make U part of the internal state. In particular, if $f : XU \to UY$ is an arrow in $\mathbf{C}$, then the isomorphism $UI \cong U$ in $\mathbf{C}$ determines a 2-cell isomorphism

$$\mathsf{Fb}^U_{X,Y}(\mathsf{Inst}(f)) \cong (U, f) : X \to Y$$

in $\mathsf{Ifo}(\mathbf{C}, \otimes)$. So the internal state of any ifo system can be viewed as the result of feeding back an instantaneous ifo system, hence the term input-feedback-output.

An important structure that the bicategory $\mathsf{Ifo}(\mathbf{C}, \otimes)$ bears is a family of arrows which we call delays. The *delay* at X is defined to be the arrow δ_X formed by feeding back X in the identity ifo system on $X \otimes X$. That is,

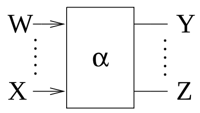
$$\delta_X = \mathsf{Fb}^X_{X,X}(I, \theta) : X \to X,$$

where $\theta : XXI \cong IXX$ is the obvious isomorphism in $\mathbf{C}$. Note that the isomorphism $XI \cong X$ in $\mathbf{C}$ determines a 2-cell isomorphism $\delta_X \cong (X, 1_{XX}) : X \to X$ in $\mathsf{Ifo}(\mathbf{C}, \otimes)$.

To each expression in the algebra of a bicategory-with-feedback there is an associated geometric presentation, along the lines of [22]. This will be treated in a future paper; now we are content to give rules for drawing a picture of any expression of ifo systems constructed from the above operations. Note that the pictures we will be drawing should only be used as an aid to representing an expression and not be used as a calculational tool. In fact, it is possible for two different (but naturally isomorphic expressions) to have the same associated picture. For the purposes of this paper, we will mean by an expression in $\mathsf{Ifo}(\mathbf{C}, \otimes)$ a well-formed word constructed from the objects and arrows of $\mathsf{Ifo}(\mathbf{C}, \otimes)$ using the above defined operations.

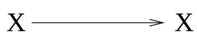
Suppose $(\mathbf{C}, \otimes)$ is a symmetric monoidal category.

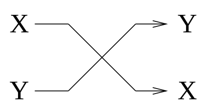
If $\alpha : W \otimes \ldots \otimes X \to Y \otimes \ldots \otimes Z$ is an ifo system in $(\mathbf{C}, \otimes)$, then we draw it as follows.



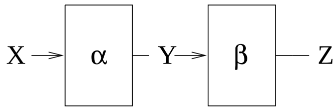
Of course, if the domain (resp. codomain) of α is the unit I for the tensor then we do not draw any wires on the left (resp. right) side of the component.

The identity ifo system $1_X : X \to X$ and the symmetry ifo system $s_{X,Y}$: $X \otimes Y \to Y \otimes X$ are drawn as follows.



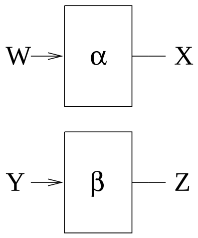


The composite of $\alpha : X \to Y$ with $\beta : Y \to Z$ is drawn as follows.

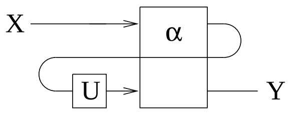


We often leave out the label on the middle wire.

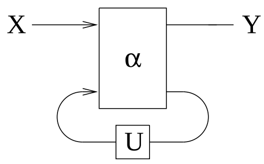
The tensor product of $\alpha : W \to X$ and $\beta : Y \to Z$ is drawn as follows.



If $\alpha : X \otimes U \to U \otimes Y$ is a an ifo system then we draw $\mathsf{Fb}^{U}_{X,Y}(\alpha) : X \to Y$ as follows.

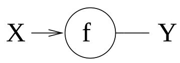


As the above picture is cumbersome to draw, we will draw a picture of another (in a sense, equivalent) expression. If $\alpha : X \otimes U \to Y \otimes U$ is an ifo system then we draw $\mathsf{Fb}^{U}_{X,Y}(s_{Y,U} \cdot \alpha)$ as follows.



The reason for the box on the feedback wire will be made clear in a later section when we compare categories-with-feedback with traced monoidal categories.

If $f : X \to Y$ is an arrow of $\mathbf{C}$ then we draw the ifo system $\mathsf{Inst}(f)$ as follows.



2.2 Examples of ifo systems

Circuits One of the original motives in [24] for considering the loops-suspension construction was to obtain a deterministic model of asynchronous circuits in which the state might be an element of an infinite data type.

We define a *circuit* in a distributive category $\mathbf{D}$ to be an ifo system in $(\mathbf{D}, \times)$. When the distributive category is Set, we define a behaviour of the circuit (U, α) :

$X \to Y$ to be an infinite sequence of triples $(x_i, u_i, y_i)_i \in (X \times U \times Y)^{\mathbb{N}}$ satisfying the difference equation $(u_{i+1}, y_i) = \alpha(x_i, u_i)$. (The object $\mathbb{N}$ is the set of natural numbers.) Think of the input x_i as an action that begins at time i and ends at time $i+1$; transforming the internal state of the circuit from u_i to u_{i+1}; and having a corresponding output y_i which also starts at time i and ends at time $i+1$. We now calculate some examples (and properties) of behaviours.

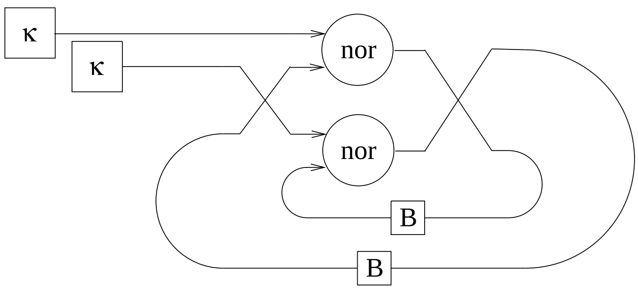
If $f : X \to Y$ is a function then the behaviour of the instantaneous ifo system $\mathsf{Inst}(f)$ is an infinite sequence of the form $(x_i, y_i)_i$ where $y_i = f(x_i)$; so an input x_i is instantaneously transformed into the output $f(x_i)$.

A behaviour for the composite of $(U, \alpha) : X \to Y$ with $(V, \beta) : Y \to Z$, is a sequence of the form $(x_i, u_i, v_i, z_i)_i$ such that $u_{i+1} = \mathrm{proj}_U \cdot \alpha(x_i, u_i)$ and $(v_{i+1}, z_i) = \beta(y_i, v_i)$ where $y_i = \mathrm{proj}_Y \cdot \alpha(x_i, u_i)$. Thus the output y_i of the first circuit serves as the input for the second circuit (from time i to time $i+1$).

If $(V, \alpha) : XU \to UY$ is a circuit then a behaviour of $\mathsf{Fb}^U_{X,Y}(V, \alpha) : X \to Y$ is a sequence of the form $(x_i, u_i, v_i, y_i)_i$ such that $(v_{i+1}, u_{i+1}, y_i) = \alpha(x_i, u_i, v_i)$. This sequence can be thought of as a behaviour of $(V, \alpha) : XU \to UY$ for which the output U is delayed by one unit of time and then (instantaneously) fed back to the input U.

A behaviour of the delay $\delta_X : X \to X$ is a sequence of triples of the form $(x_{i+1}, x_i, x_i)_{i \in \mathbb{N}}$. As the output sequence of such a delay comprises the input sequence delayed by one unit of time (plus an initial output), it would be more precise to call such a component a unit delay.

We now model a pair of nor gates fed back to one another. Let $B = \{0, 1\}$, let nor $: B \times B \to B$ be the function defined by a nor $b =$ not(a or b), let $\Delta : B \to B \times B$ be the diagonal, and let κ be the circuit $(B, \Delta) : I \to B$. Consider the following circuit.

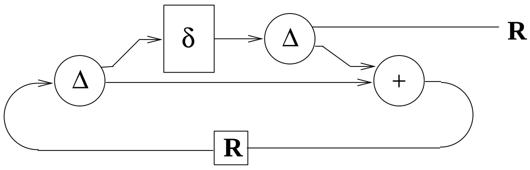


It is straightforward to check that a behaviour of this circuit is a sequence $(x_i, y_i, u_i, v_i)_{i \in \mathbb{N}} \in (B^4)^{\mathbb{N}}$ satisfying

$$(x_{i+1}, y_{i+1}, u_{i+1}, v_{i+1}) = (x_i, y_i, x_i \text{ nor } v_i, y_i \text{ nor } u_i).$$

Here the variables x and y are the internal states (as well as constant outputs) of the two κ circuits and the variables u and v are the internal states resulting from the two instances feedback.

As a final example, we construct a circuit which outputs the Fibonacci numbers. Let $+ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be addition on the reals, let $\Delta : \mathbb{R} \to \mathbb{R} \times \mathbb{R}$ be the diagonal function and let δ be the delay circuit at $\mathbb{R}$. Define $F : I \to \mathbb{R}$ to be the following circuit.



Notice that F has internal state $\mathbb{R} \times \mathbb{R}$ and its output behaviour, when started in state $(1, 1) \in \mathbb{R} \times \mathbb{R}$, will give the Fibonacci numbers. (More precisely, the output from time i to time $i + 1$ will be the $(i + 1)^{\text{th}}$ Fibonacci number.)

Of course, different notions of behaviour are more appropriate in other circumstances. For example, we define an input-output behaviour of $(U, \alpha) : X \to Y$ to be a sequence $(x_i, y_i)_{i\in\mathbb{N}}$ such that there exists an infinite sequence of triples $(x_i, u_i, y_i)_i \in (X \times U \times Y)^{\mathbb{N}}$ such that $(u_{i+1}, y_i) = \alpha(x_i, u_i)$. It is clear that the input-output behaviours of $(U, \alpha) : X \to Y$ comprise a relation $\mathsf{i\text{-}o}(U, \alpha) : X^{\mathbb{N}} \to Y^{\mathbb{N}}$. If F is the Fibonacci circuit described above then $\mathsf{i\text{-}o}(F) : I \to \mathbb{R}^{\mathbb{N}}$ is the subset of sequences $(r_i)_{i\in\mathbb{N}}$ satisfying, for all $i \in \mathbb{N}$, the recurrence relation $r_{i+2} = r_{i+1} + r_i$. In Section 2.3 we will see that $\mathsf{i\text{-}o}$ defines a homomorphism of bicategories.

Relations of the form $X^{\mathbb{N}} \to Y^{\mathbb{N}}$ provide a useful calculus for specifying (input-output behaviours of) circuits. In fact, in [8] and [20] relations were used in this way to model circuits. Note, however, that the input-output behaviours of circuits with input and output I (such as the example above involving the nor function) are trivial – there are only two relations from $I^{\mathbb{N}}$ to $I^{\mathbb{N}}$. More will be said on behaviours in the next section.

In [49] it was shown that distributive categories provide a setting for defining data-types; they also have the required structures for modelling asynchronous circuits as ifo systems. In [52] and [28], different specifications on the data types and arrows of $\mathsf{Ifo}(\mathsf{Set}, \times)$ are used to give certain subcategories of ifo systems which capture different paradigms of communication between circuit components. In particular, the notion of a *pulse circuit* is used for constructing asynchronous circuits for computing recursive functions as well as asynchronous pipelines (abstracted from Sutherland's *micropipeline* [45]).

Algorithms The other original motivation for considering ifo systems was to give an algebra of algorithms in a distributive category of data types.

We define an *Elgot automaton* in a distributive category $\mathbf{D}$ to be an ifo system in $(\mathbf{D}, +)$. Generally, a behaviour of an Elgot automaton $(U, \alpha) : X \to Y$ in Set is taken to be a possibly infinite sequence $x, u_0, u_1, \ldots$ satisfying $u_0 = \alpha(x)$ and $u_{i+1} = \alpha(u_i)$. (Of course, the sequence terminates if for some n, $\alpha(u_n) \in Y$.) A more abstracted notion of the behaviour of this Elgot automaton is a partial function $f : X \rightharpoonup Y \times \mathbb{N}$ where $f(x) = (y, n)$ if the the above behaviour sequence

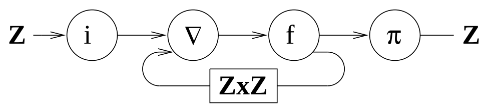
$x, u_0, u_1, \ldots$ terminates with $u_n = y$. (We call such a structure a partial function with duration.) One could abstract away the temporal aspect altogether, and consider the partial function $g : X \rightharpoonup Y$ which describes only the result of the computation of the automaton (so $g(x)$ is defined if there exists an n such that the sequence $x, u_0, u_1, \ldots$ terminates with $u_n = g(x) \in Y$).

The notions of input and output for an Elgot automaton are different to those for circuits; an input in this case is an initial state (rather than an action) and an output is a terminal state. As was pointed out in [24], the algebra of Elgot automata allows one to decompose the state-space of a system (in that it uses coproducts) while the algebra of circuits allows one to decompose the underlying space of a system (in that it uses products). Moreover, the picture of an expression of Elgot automata should be thought of as a flow chart with feedback, while the picture of an expression of circuits should be thought of as a circuit diagram.

We now construct an Elgot automaton which computes factorial. Consider the function

$$\begin{aligned} f &: \mathbb{Z}^2 \to \mathbb{Z}^2 + \mathbb{Z}^2 \\ &: (x,y) \mapsto \begin{cases} \mathrm{inj}_1(xy, y-1) \text{ if } y = 1 \\ \mathrm{inj}_2(xy, y-1) \text{ otherwise} \end{cases} \end{aligned}$$

So $f(x,y)$ lies in the first summand of $\mathbb{Z}^2 + \mathbb{Z}^2$ if and only if $y = 1$. Let $\nabla : \mathbb{Z}^2 + \mathbb{Z}^2 \to \mathbb{Z}^2$ be the codiagonal, let $i : \mathbb{Z} \to \mathbb{Z}^2$ be the function sending r to $i(r) = (1, r)$ and let $\pi : \mathbb{Z}^2 \to \mathbb{Z}$ be the projection onto the first variable. Below we construct an Elgot automaton which computes the partial function $\alpha : \mathbb{Z} \rightharpoonup \mathbb{Z}$ satisfying $\alpha(r)$ (is defined if and only if r is a positive integer and in which case) equals $r!$.



Suppose $\mathbf{D}$ is initial amongst distributive categories which come equipped with a fixed point N of the functor $(-) + I$ (see [16] for a description of this category) – the object N is interpreted as the data type of the natural numbers with predecessor and successor; then the algebra of Elgot automata in $\mathbf{D}$ can be used to obtain the classical computation of recursive functions [42]. But further, when instead the data type of real numbers is taken [48], with its operations as an ordered ring, the functions computable on the reals, in the sense of Blum, Shub and Smale, [7] are obtained.

There is a second tensor product $\otimes$ on $\mathsf{Ifo}(\mathbf{D}, +)$ which is inherited from the cartesian product of $\mathbf{D}$. It is, in fact, a lax tensor product on the bicategory $\mathsf{Ifo}(\mathbf{D}, +)$ in the sense that, if (f, g) and (h, i) are both composable pairs of ifo systems, there is a (not necessarily invertible) 2-cell $(g \otimes i) \cdot (f \otimes h) \Rightarrow (g \cdot f) \otimes (i \cdot h)$. This models the important operation of computing functions asynchronously and synchronizing on completion.

Finite State Automata In [6] a 2-category of nondeterministic finite state automata over an alphabet A is described, including a behaviour functor to matrices of languages, and an account of the Kleene theorem, and minimal realization. This construction turns out to be a special case of loops-suspension, namely that applied to the additive category of matrices on the semiring of subsets of A*, with direct sum as the tensor product involved.

Linear system theory The bicategory $\mathsf{Ifo}(\mathsf{Vect}, \oplus)$, gives an algebra of discrete linear systems; that is input-output systems governed by a system of linear recurrence relations with constant coefficients ([40]). What is more surprising is that the same category can be used to model continuous linear systems, as will be described in a later section.

Mealy Automata An ifo system in $(\mathsf{Set}_{\text{finite}}, \times)$ – that is, a finite circuit – is just a deterministic Mealy automaton ([43]). Further, an ifo system in the category of relations $(\mathsf{Rel}, \otimes)$ is a non-deterministic Mealy automaton. So ifo systems also provide an algebra for Mealy automata.

2.3 Behaviour

In order to study bicategories of ifo systems, we often consider structure-preserving homomorphisms to other bicategories. Often such homomorphisms deserve the name *behaviour.*

Given any finitely complete category $\mathbf{E}$, recall that the bicategory $\mathbf{Span}(\mathbf{E})$ of spans ([4]) in $\mathbf{E}$ is a self-dual compact closed bicategory (a symmetric monoidal autonomous bicategory, in the language of [14]), the tensor product $\otimes$ being the obvious one inherited from the Cartesian structure of $\mathbf{E}$. It thus admits a feedback operation or a trace. (It fails to be a traced monoidal category only because it is a bicategory rather than a category.) Explicitly, the trace is defined as follows. Consider the following span α.

$$\begin{tikzcd} & W \arrow[dl, "f"'] \arrow[dr, "g"] & \\ XY & & YZ \end{tikzcd}$$

Then $\mathsf{Tr}^Y_{X,Z}(\alpha)$ is the span

$$\begin{tikzcd} & E \arrow[dl, "\mathrm{proj}_X \cdot e"'] \arrow[dr, "\mathrm{proj}_Z \cdot e"] & \\ X & & Z \end{tikzcd}$$

where $e : E \to W$ is the equalizer of the two arrows $\mathrm{proj}_Y \cdot f$ and $\mathrm{proj}_Y \cdot g$. This trace operation is defined on the 2-cells of $\mathbf{Span}(\mathbf{E})$ in the obvious way. For any triple of objects X, Y and Z, this defines a functor

$$\mathsf{Tr}^Y_{X,Z} : \mathbf{Span}(\mathbf{E})(XY, YZ) \to \mathbf{Span}(\mathbf{E})(X, Z).$$

We now define a strong monoidal homomorphism

$$\mathcal{B} : \mathsf{Ifo}(\mathsf{Set}, \times) \to (\mathbf{Span}(\mathsf{Set}), \otimes).$$

It maps an object X to the set $X^{\mathbb{N}}$ and, if $(U, \alpha) : X \to Y$ is a circuit, $\mathcal{B}(U, \alpha)$ is the span of sets $X^{\mathbb{N}} \leftarrow W \to Y^{\mathbb{N}}$ where $W = \{(x_i, u_i, y_i)_{i \in \mathbb{N}} \mid \alpha(x_i, u_i) = (u_{i+1}, y_i)\}$. So an element of $\mathcal{B}(U, \alpha)$ is a behaviour of (U, α) as described in the previous subsection on circuits.

It is straightforward to check that the homomorphism $\mathcal{B}$ does not preserve feedback. This fact will be remarked on in Section 3 wherein we will see that $\mathcal{B}$ can be factored through a homomorphism

$$\mathcal{D} : \mathsf{Ifo}(\mathsf{Set}, \times) \to \mathbf{Span}(\mathbf{Graph}(\mathsf{Set})).$$

By composing $\mathcal{B}$ with the homomorphism $\mathbf{Span}(\mathsf{Set}) \to \mathsf{Rel}$ to the bicategory of relations (which is the universal homomorphism from a bicategory to a locally ordered category), we obtain another behaviour homomorphism

$$\mathsf{i\text{-}o} : \mathsf{Ifo}(\mathsf{Set}, \times) \to \mathsf{Rel}$$

which only records the input-output aspect of the behaviour of a circuit (as described in the earlier section on circuits).

We often refer to the domain of a behaviour homomorphism as the machine bicategory (for the homomorphism); and the codomain as the behaviour bicategory. Behaviour bicategories tend to provide more abstract models of dynamical systems than the corresponding machine bicategories. For example, a span of sets from $X^{\mathbb{N}}$ to $Y^{\mathbb{N}}$ does not explicitly model the internal dynamics of a system as does a circuit from X to Y; and a relation from $X^{\mathbb{N}}$ to $Y^{\mathbb{N}}$ describes neither the internal dynamics nor the internal state.

Often we abstract away more than just the dynamics and internal state. In the next subsection examples of behaviour functors will be given which also abstract away the temporal aspects of ifo systems.

2.4 Categories-with-feedback and traced monoidal categories

One of the basic notions underlying the work presented in this paper is that of a bicategory-with-feedback – a concept we have not defined. In order to avoid burdening the reader with too many the bicategorical details and to make a precise connection with traced monoidal categories, we will only define the notion of a category-with-feedback. This will be done by using the following construction.

Given a bicategory $\mathcal{B}$, one can form the category whose objects are those of $\mathcal{B}$ and in which an arrow is an isomorphism class of arrows in $\mathcal{B}$. Identities and composition are inherited from $\mathcal{B}$ in the obvious way. We call this category the *quotient category* of $\mathcal{B}$, and we denote it by $\rfloor\mathcal{B}\lfloor$. This construction defines a functor $\rfloor - \lfloor : \mathsf{Bicat} \to \mathsf{Cat}$, where Bicat is the category of bicategories and homomorphisms and Cat is the category of categories and functors.

This construction can be found in [4]. It is also used in the definition of the Grothendieck group of a monoidal category: the Grothendieck group of a monoidal category $(\mathbf{C}, \otimes)$ is formed by inverting the arrows of $\rfloor \Sigma(\mathbf{C}, \otimes) \lfloor$.

Many structures that a bicategory may bear can be passed on to its quotient category. For example, if $\mathcal{B}$ is a symmetric monoidal (resp. compact closed) bicategory then $\rfloor \mathcal{B} \lfloor$ naturally inherits the structure of a symmetric monoidal (resp. compact closed) category. Moreover, we have the functor

$$\rfloor - \lfloor : \mathsf{SymMonBicat} \to \mathsf{SymMonCat}.$$

The functor

$$\mathcal{M} = \rfloor \mathsf{Ifo}(-) \lfloor : \mathsf{SymMonCat} \to \mathsf{SymMonCat}$$

naturally admits the structure of a monad, which we now describe. The symmetric strong monoidal functors

$$\rfloor \mathsf{Inst} \lfloor : (\mathbf{C}, \otimes) = \rfloor (\mathbf{C}, \otimes) \lfloor \to \mathcal{M}(\mathbf{C}, \otimes)$$

comprise the unit for the monad. The multiplication for the monad comprises symmetric strong monoidal functors denoted

$$\mathsf{Fb} : \mathcal{M}^2(\mathbf{C}, \otimes) \to \mathcal{M}(\mathbf{C}, \otimes)$$

which will now be defined in terms of the feedback operations $\mathsf{Fb}^{Y}_{X,Z}$ of ifo systems. First, if f is an arrow of a bicategory $\mathcal{B}$ let us denote by $[f]$ the arrow of $\rfloor \mathcal{B} \lfloor$ which is the equivalence class containing f. On objects, Fb is the identity function. Now, suppose $[(Y, [(U, \alpha)])] : X \to Z$ is an arrow of $\mathcal{M}^2(\mathbf{C}, \otimes)$. So α is a morphism $XYU \to UYZ$ in $\mathbf{C}$. We define $\mathsf{Fb}([(Y, [(U, \alpha)])])$ to be $[\mathsf{Fb}^{Y}_{X,Z}(U, \alpha)]$. Note that the functors Inst and Fb are, in fact, strict monoidal.

We define a *category-with-feedback* to be an algebra $F : \mathcal{M}(\mathbf{E}, \otimes) \to (\mathbf{E}, \otimes)$ for this monad. If α is an arrow of $\mathbf{E}$ of the form $XU \to UY$, we denote $F([(U, \alpha)])$ by $F^{U}_{X,Y}(\alpha)$. For each object $X \in \mathbf{E}$, we define the *delay* at X to be the arrow $F^{X}_{X,X}(1_{X \otimes X})$. Note that the arrows of $\mathcal{M}(\mathbf{C}, \otimes)$, the free category-with-feedback on $(\mathbf{C}, \otimes)$, are equivalence classes of ifo systems. We draw pictures for expressions in categories-with-feedback in the same way as we did for ifo systems.

A traced monoidal category [22] is a balanced monoidal category $(\mathbf{E}, \otimes)$ equipped with a family of functions $\mathsf{Tr}^{U}_{X,Y} : \mathbf{E}(X \otimes U, Y \otimes U) \to \mathbf{E}(X, Y)$ satisfying a number of axioms. Of course, in the presence of the symmetry, one could equally give functions of the form $\mathbf{E}(X \otimes U, U \otimes Y) \to \mathbf{E}(X, Y)$. Earlier we noted that the trace Tr of spans fails to satisfy the axioms for a trace only because spans constitute a bicategory rather than a category. It is easy to show that, for any finitely complete $\mathbf{B}$, the trace for spans equips $\rfloor \mathbf{Span}(\mathbf{B}) \lfloor$ with a traced monoidal structure. (This is, of course, equal to the traced monoidal structure constructed from the compact closed structure of $\rfloor \mathbf{Span}(\mathbf{B}) \lfloor$.)

The following theorem identifies traced symmetric monoidal categories as a class of categories-with-feedback. The proof, which is otherwise straightforward, uses Lemma 2.1 of this last mentioned paper, which states that slidings of crossings and twists suffice for all slidings in the presence of the other traced monoidal axioms.

Theorem 1. *A traced symmetric monoidal category is a category-with-feedback where the delay at each object is the identity.*

This theorem may be interpreted as saying that a trace is an instantaneous feedback. When drawing the feedback operation in a traced symmetric monoidal category (viewed as a category-with-feedback) we leave the box off the fed-back wire. The resulting diagrams are then equivalent to the string diagrams of [22] (modulo the fact that we are only dealing with the symmetric case).

From the point of view of the geometry of tensor calculus it seems that traced monoidal categories provide a more natural theory. The existence of the delays, however, is intimately related to the notion of internal state and is critical for our theory: in order to describe the process (as well as the effect) of computation, we need to model the evolution of the internal state of a system.

Of course, the above theorem allows us to construct the free traced monoidal category on a symmetric monoidal category: first form the category of ifo systems and then force the delays to be identities.

We are not proposing this construction as a means for calculating in general free traced monoidal categories – the results of [22] give a geometric calculus of string diagrams which enables one to do this. Rather, the above construction provides a means whereby one may obtain a non-syntactic description of the free traced monoidal category on a symmetric monoidal category. In general, of course, quotients of monoidal categories, such as the one given above, are difficult to compute.

It is not too difficult to show that the free traced monoidal category on $(\mathsf{Set}_{\mathrm{finite}}, +)$ is the category of 'partial functions with possibly several undefined values' – that is, cospans $X \to W \leftarrow Y$ in $\mathsf{Set}_{\mathrm{finite}}$ where the arrow $Y \to W$ is an inclusion. (In order to perform this calculation one uses the representation of Elgot automata as cospans of graphs which is described in the third last paragraph of Section 3.1. Forcing delays to be identities then corresponds to collapsing the edges of graphs – that is, collapsing a graph to its discrete graph of connected components.)

It is possible to extend the construction $\mathcal{M}$ from symmetric to braided monoidal categories. We will not pursue this direction as our present interest is in the algebraic foundations of dynamical systems rather than the geometry of tensor calculus; however, we do mention that (as one would expect) there is a close connection between the free traced monoidal category on the category of braids ([21]) and categories of tangles ([15]).

2.5 Equilibrium behaviours

When studying dynamical systems, we often want to consider 'equilibrium' behaviours. We call a morphism H of categories-with-feedback an *equilibrium* functor if its codomain is a traced monoidal category. Notice that H maps delays to identities. The construction of free traced monoidal categories provides examples of such functors. Equilibrium functors may also be constructed in the following way.

Suppose $(\mathbf{C}, \otimes)$ is a symmetric monoidal category and let $\mathsf{Tr} : \mathcal{M}(\mathbf{E}, \otimes) \to (\mathbf{E}, \otimes)$ be a traced symmetric monoidal category. Given a strong symmetric monoidal functor $G : \mathbf{C} \to \mathbf{E}$, we can form the composite functor

$$\mathsf{Tr} \cdot \mathcal{M}(G) : \mathcal{M}(\mathbf{C}, \otimes) \to \mathbf{E}$$

which is a morphism of categories-with-feedback. (Note this is an instance of the adjointness of free categories-with-feedback where the codomain happens to be traced monoidal.)

For example, take the functor G to be the inclusion of $(\mathsf{Set}, \times)$ into the category $(\mathsf{Rel}, \otimes)$ of relations. Then the corresponding functor

$$\mathcal{M}(\mathsf{Set}, \times) \to \mathsf{Rel}$$

maps (the equivalence class of) a circuit $(U, \alpha) : X \to Y$ to the relation $R : X \to Y$ where xRy if and only if there exists $u \in U$ such that $\alpha(x, u) = (u, y)$. Such an internal state u is called an *equilibrium* state for input x and output y. This equilibrium functor into Rel has 'eliminated' time, unlike the homomorphism $\mathsf{i\text{-}o} : \mathsf{Ifo}(\mathbf{Set}, \times) \to \mathsf{Rel}$ defined at the end of Section 2.3, in the sense that it maps delays to identities. Clearly it gives a more abstract measure of a circuit.

As another example, take the functor G to be the inclusion of $(\mathsf{Set}, +)$ into the category $(\mathsf{Par}, +)$ of partial functions. The resulting functor

$$\mathcal{M}(\mathsf{Set}, +) \to \mathsf{Par}$$

maps an Elgot automaton $(U, \alpha) : X \to Y$ to the partial function $f : X \rightharpoonup Y$ which describes the result of the computation of the Elgot automaton (as defined in the earlier section on algorithms). We can use the fact that this functor preserves composition, tensor and feedback when calculating the partial function computed by an expression of Elgot automata. For example, it is straightforward to verify that the partial function of the Elgot automaton constructed earlier for computing factorial satisfies the required fixed point equations for factorial.

In circuit theory, an important notion is that of a stabilizing circuit: if an input is held fixed then after some time the internal state will stabilize, as will the output. Further work is needed in order to express this notion of equilibrium as a combination of (or in a way related to) these last two functors.

3 Span(Graph)

In this section we introduce the bicategory of spans of graphs and describe its relationship to categories of ifo systems (in categories with products) via a homomorphism $\mathcal{D}$. We further indicate how it provides an algebra for labelled transition systems, thus serving as a tool for specifying concurrent systems.

3.1 Systems with boundaries

Let $\mathbf{Graph}(\mathbf{C})$ denote the category of directed graphs in $\mathbf{C}$. We will denote a directed graph as a pair of arrows $s, t : E \to V$ or, if $\mathbf{C}$ has products, as an arrow of the form $(s,t) : E \to V^2$. A graph is often thought of as a dynamical system in $\mathbf{C}$: the object of edges E represent motions of a system, the object of vertices V represent states and the source and target maps s and t give the starting and finishing states of a motion. Deterministic dynamical systems are graphs of the form $1_V, s : V \to V$; that is, they are endomorphisms.

If $\mathbf{C}$ is a finitely complete category let $\mathbf{Span}(\mathbf{Graph}(\mathbf{C}))$ denote the bicategory of spans of graphs in $\mathbf{C}$. In this section, we will only consider its full and locally full sub-bicategory whose objects are 'one vertex' graphs – that is, graphs whose object V of vertices is terminal in $\mathbf{C}$. We will denote this bicategory by $\mathbf{SpGr}(\mathbf{C})$.

Note that $\mathbf{SpGr}(\mathbf{C})$ is a sub-compact-closed-bicategory of $\mathbf{Span}(\mathbf{Graph}(\mathbf{C}))$ and thus admits a traced monoidal structure. When drawing pictures of expressions in $\mathbf{SpGr}(\mathbf{C})$ we will leave the arrow heads off the input wires of components.

Given a span $A \leftarrow G \to B$, we call the graph G the *head* or *centre* of the span and the two morphisms are named the left and right legs. The *opposite* of a span is formed by swapping the left and right legs. (This is the dual construction available in any compact closed category.)

We think of such a span as modelling the states and transitions of a system with two specified parts, the legs of the span restricting the state and motion of the whole to these parts. The operation of composition of spans is formed by a pullback; and it corresponds to glueing two systems together on the parts of the system corresponding to the legs of the span involved in the pullback. (Of course, a more detailed model of such a system would include a description of its underlying space as well as its state-space – the latter being thought of as a space of quantities varying over the former. The underlying space could be modelled as a cospan, the legs of the cospan including the specified parts; and the composite – or glueing – of two systems could be modelled by composition of cospans, that is, by a pushout. At the level of the state-space, the cospans become spans and the pushouts become pullbacks.)

If A and B are one vertex graphs – as in $\mathbf{SpGr}(\mathbf{C})$ – then, on these parts, we are recording information pertaining only to the motion of the system and not the state. In this sense, we are thinking of these parts as *boundaries* of the system across which we are recording 'pure motion'. (In Section 4, we indicate how this algebra can be used to model continuous systems. At the end of Section 5, we model systems governed by the continuity equation as certain spans, the legs of these spans recording flows across boundaries.)

Given a finitely complete $\mathbf{C}$ with terminal object 1, let

$$\mathcal{D} : \mathsf{Ifo}(\mathbf{C}, \times) \to \mathbf{SpGr}(\mathbf{C})$$

be the homomorphism specified as follows. The object X is mapped to the one vertex graph $(!, !) : X \to 1^2$; we often denote this graph by X. Let $(U, \alpha) : X \to Y$ be an ifo system where $\alpha = (\beta, \gamma) : XU \to UY$. Then $\mathcal{D}(U, \alpha)$ has as centre the graph $(\mathrm{proj}_2, \beta) : XU \to U^2$, as left leg the map $\mathrm{proj}_1 : XU \to X$ and as right leg the map $\gamma : XU \to Y$. It is straightforward to check that $\mathcal{D}$ is a strong monoidal homomorphism which identifies $\mathsf{Ifo}(\mathbf{C}, \times)$ as a sub-monoidal bicategory of $\mathbf{SpGr}(\mathbf{C})$. We call the arrows of this sub-bicategory *deterministic input-driven* systems in $\mathbf{C}$. Deterministic input-driven systems can be characterized (for appropriate categories $\mathbf{C}$) as spans of graphs whose left leg satisfies the discrete op-fibration condition. (This observation relates the work at hand to [33] and [34]; the relation of spans and cospans of categories to ifo and other input-output systems was studied in detail in [23].)

Notice that if we feedback an input-driven system in $\mathbf{SpGr}(\mathbf{C})$ using the trace, we do not necessarily obtain another input-driven system. For example, in the case $\mathbf{C} = \mathsf{Set}$, consider $\mathsf{Tr}^B_{1,1}(\mathcal{D}(\mathsf{Inst}(\mathrm{not}))) : 1 \to 1$, where $B = \{ \text{ true }, \text{ false } \}$ and $\mathrm{not} : 1 \times B \cong B \to B \cong B \times 1$ is the usual negation function composed with the obvious isomorphisms.

However, the homomorphism $\mathcal{D}$ does satisfy the property

$$\mathcal{D}(\mathsf{Fb}^U_{X,Y}(f)) \cong \mathsf{Tr}^U_{X,Y}(\mathcal{D}(f \cdot (X\delta_U))),$$

where $f : XU \to UY$ is an ifo system and δ_U is the delay at U.

After Theorem 1, we remarked that a trace is an instantaneous feedback. In view of the above stated property of $\mathcal{D}$, it would be reasonable to say that the feedback of ifo systems is a delayed trace. (The feedback $\mathsf{Tr}^U_{X,Y}$ in $\mathbf{Span}$ should be thought of as glueing the two boundaries whose state-spaces are both U, thus forcing the relevant transitions to be the same. Of course, it is common practice in Engineering to delay feedback because of the problems that can arise from such a glueing operation.)

Consider the case $(\mathbf{C}, \otimes) = (\mathsf{Set}, \times)$. Let N be the graph $1_{\mathbb{N}}, s : \mathbb{N} \to \mathbb{N}$ where s is the successor function on the set of natural numbers. We think of a graph morphism $N \to G$ as a behaviour of G. As the functor $\mathcal{F} = \mathbf{Graph}(N, -) : \mathbf{Graph}(\mathsf{Set}) \to \mathsf{Set}$ preserves limits, we can consider the homomorphism of bicategories

$$\mathbf{Span}(\mathcal{F}) : \mathbf{SpGr}(\mathsf{Set}) \hookrightarrow \mathbf{Span}(\mathbf{Graph}(\mathsf{Set})) \to \mathbf{Span}(\mathsf{Set}).$$

The homomorphism $\mathcal{B} : \mathsf{Ifo}(\mathsf{Set}, \times) \to (\mathbf{Span}(\mathsf{Set}), \otimes)$ described in Section 2.3 can be factored as the composite $\mathbf{Span}(\mathcal{F}) \cdot \mathcal{D}$.

As was made clear in [35], an extra structure which categories of dynamical systems (in which the *objects* are models of systems) should bear is a *time object* T (or a collection of such objects) which should serve as a domain for behaviour in the sense that a T-behaviour of a system G should be a map $T \to G$. For example, the above graph N is usually taken to be the time object in $\mathbf{Graph}(\mathsf{Set})$. Of course, once given a time object, behaviour functors can be formed as above. In Section 4.3 we will see an example of a time object which parametrizes behaviours of continuous systems.

It is possible to extend the construction Ifo from monoidal categories to monoidal bicategories. (In order to obtain the desired 2-cells, we need to consider equipments [12], [11] rather than bicategories. Examples of such equipments are given by the homomorphisms Inst.) We mention this in order to note that the sub-bicategory $\mathbf{SpGr}(\mathbf{C})$ of $\mathbf{Span}(\mathbf{Graph}(\mathbf{C}))$ is biequivalent to $\mathsf{Ifo}(\mathbf{Span}(\mathbf{C}), \otimes)$ and thus also fits into the context of ifo systems. The reader can check that to give an ifo system $(U, \alpha) : X \to Y$ in $\mathbf{Span}(\mathbf{C})$ is just to give a span of graphs in $\mathbf{C}$ with domain $(!, !) : X \to 1^2$ and codomain $(!, !) : Y \to 1^2$.

Furthermore, the above homomorphism $\mathcal{D}$ can be constructed as $\mathsf{Ifo}(\mathcal{I})$, where $\mathcal{I} : (\mathbf{C}, \times) \to (\mathbf{Span}(\mathbf{C}), \otimes)$ is the obvious inclusion. Similarly, Elgot automata can be represented as cospans of graphs by realizing $\mathsf{Ifo}(\mathbf{Cospan}(\mathbf{C}), +)$ as a sub-bicategory of $\mathbf{Cospan}(\mathbf{Graph}(\mathbf{C}))$.

In the process of designing and analysing (deterministic) systems it is often necessary to consider operations which yield non-deterministic systems and this is why we are interested in the passage from ifo systems to spans. For example, the opposite of the diagonal $\Delta^* : X \otimes X \to X$ and the opposite of the terminal $!^* : I \to X$ play an important role in the design of systems though they are not ifo systems. In Section 4 we will see that in order to analyse (in fact, to construct) RLC circuits we need to pass from the algebra of ifo systems to the compact closed bicategory $\mathbf{SpGr}(\mathsf{Vect})$.

Deterministic systems tend to have an infinite state-space, but when designing systems we often deal with finite abstractions. It is worth mentioning that the word 'abstraction' has two meanings here: it can refer to the process of abstraction or it can refer to the result of this process. This process of abstraction can be described (via 2-cells) in $\mathbf{SpGr}(\mathsf{Set})$; in fact, we can define an algebra of abstractions. We will not discuss this further other than saying that it is closely related to the constructions presented in Section 5. With respect to the second meaning of abstraction, we will now demonstrate how spans of finite and reflexive graphs provide an algebra for specifying concurrent systems.

3.2 Transition systems

The original motivation for considering $\mathbf{Span}(\mathbf{Graph}(\mathsf{Set}))$ in [25] was to have a compositional algebra of finite transition systems [3]. The algebra and its associated geometry allow the consideration of spatial distribution of the parts of a system. One of the first things we observed is that there is a class of transition systems, which we called linearizable, for which interleaving semantics is reasonable, and for these systems we defined the analogue of the Hoare parallel operation, thus establishing a clear connection with Hoare's CSP. It was also straightforward to give a compositional account of P/T nets in $\mathbf{Span}(\mathbf{Graph})$ [26], where the geometry is the usual geometry of nets.

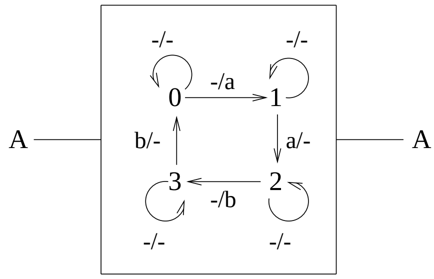
Of course, transition systems have a wide variety of applications, and the geometry we obtain fits in very naturally with these applications. For example, it is possible to model in $\mathbf{Span}(\mathbf{Graph})$ ([52], [29]) non-deterministic asynchronous circuits with finite state (for which the associated geometry yields the usual circuit diagrams) obtaining a compositional version of the general multiple winner model described in Brzozowski and Seger [9]. In [18], an interpretation of the stubborn set model checking technique ([47], [46]) is given as well as a model of communication protocols for which the associated geometry is similar to the flow charts of [19]. In addition, new applications were suggested by our algebra, one [27] being a formalization of double-entry accounting, which we describe in Section 5. Below we provide a brief introduction to $\mathbf{SpGr}(\mathsf{Set})$ as an algebra for specifying concurrent systems.

In this section we will only be concerned with the sub-bicategory of $\mathbf{SpGr}(\mathsf{Set})$ comprising spans of finite, reflexive graphs (where we also require the legs of the spans to be reflexive graph morphisms). If A is a finite set, then denote by A^+ the one-vertex reflexive graph with edge-set $A + \{\text{-}\}$, the symbol - representing the reflexive edge. So a span of reflexive graphs from A^+ to B^+ can be thought of as a labelled transition system with labels in the set $(A + \{\text{-}\}) \times (B + \{\text{-}\}) \cong A \times B + A + B + 1$. More generally, a span of reflexive graphs whose domain and codomain are products of one vertex graphs can be viewed as a transition system labelled in a 'distributed' alphabet.

A span $A^+ \leftarrow G \rightarrow B^+$ of reflexive graphs is called *linear* if no edge of G is labelled by both A and B; that is, the label of each edge lands in the subset $A + B + 1 \subset (A + \{\text{-}\}) \times (B + \{\text{-}\})$.

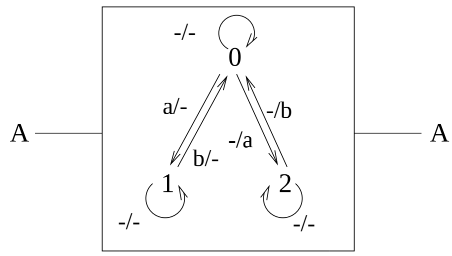
As an example, let us consider the simple distributed resource allocation problem. A group of n people go to a Chinese restaurant for dinner. After being sat down at a round table, they are told there is a resource problem: there are only n chopsticks. The diners decide to share the chopsticks (placing one between each adjacent pair of diners), and each diner agrees to obey the following protocol: when a diner decides he wants to eat, he must first pick up his right chopstick and then pick up his left; and when he has eaten enough, he must first put down his right chopstick and then put down his left. Note that a diner is not allowed to put down his right chopstick until he has picked up his left. This resource allocation protocol is distributed because there is no global control of the use of the chopsticks – each diner (and each chopstick) acts independently of the other diners (and the other chopsticks).

In $\mathbf{SpGr}(\mathsf{Set})$ we model diners and chopsticks by linear spans of reflexive graphs, and the whole system is modelled by an expression involving these spans. Let $A = \{a, b\}$, and let $\beta : A^+ \to A^+$ be the span of graphs depicted below.



This span models a diner: the vertex 0 corresponds to the diner having no chopsticks and the reflexive edge -/- at 0 corresponds to a transition in which the diner does nothing (perhaps he is waiting to take his right chopstick or perhaps he is busy conversing); the transition labelled -/a from 0 to 1 corresponds to the diner picking up his right chopstick; the next non-reflexive transition a/- corresponds to the diner picking up his left chopstick; and the transitions -/b and b/- respectively correspond to the diner putting down his right and his left chopsticks.

A chopstick is modelled by the span $\chi : A^+ \to A^+$ depicted below.

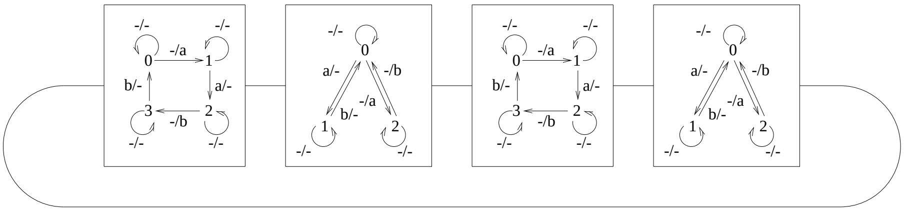


The vertex 0 corresponds to the chopstick being on the table; vertex 1 corresponds to it being used by the diner on its left; and vertex 2 corresponds to it being used by the diner on the its right.

The system of n *diners and* n *chopsticks* is defined to be the expression

$$\mathsf{Tr}_{I,I}^{A^+}((\chi \cdot \beta)^n).$$

The operations of composition and trace (which are constructed using pullbacks in the category of graphs) provide the required synchronization between the various components. For example, there is a behaviour of two diners and two chopsticks



(recall that such a behaviour is an infinite path in the graph $\mathsf{Tr}^{A^+}_{I,I}((\chi\cdot\beta)^2))$ which has the following sequence of states.

$$(0,0,0,0),(1,1,0,0),(2,1,0,2),(3,0,0,2),(0,0,0,0),(0,0,0,0),(1,1,1,1),\ldots$$

Notice that the system cannot leave the vertex $(1,1,1,1)$. This is called a deadlock state and such vertices will be discussed in the next section. Though the above system is small, the relevance of the reflexive-edges for modelling concurrency should be apparent; for example, in the above behaviour the second diner was able to wait for a turn to get his right chopstick. Also, it is clear that such a system comes equipped with an initial state – for example, our interpretation would not make sense if we were to start the two diner system in state $(1,0,1,0)$.

The basic components of the diner system (namely, β and χ) are linear. It should be clear, however, that the composite of linear systems is not linear; if it were, the trace involved in the above example would not have enabled the last chopstick to synchronize with the first diner. However, the composite of linear systems is *linearizable* (a concept defined in [25]) – and it is this fact which explains the applicability of interleaving models of concurrency.

4 Continuous linear system theory

The mathematical structure of a graph can, in some cases, model continuous systems: the passive and active restrictions of a graph can be thought of as giving the position p and velocity v of an infinitesimal motion rather than source and target of a discrete motion. In other words, in an appropriate setting, a graph $(p,v): M \to S^2$ can model a system with state-space S and a space of infinitesimal motions M. Of course, this interpretation of a graph is only meaningful if the tangent bundle of the space S is isomorphic to S^2.

In this section we will see that an input-output system governed by a system of linear differential equations [40] can be considered as an ifo system in $(\mathsf{Vect},\oplus)$. There are two homomorphisms $\mathcal{D}$ and $\mathcal{I}$ to $\mathbf{SpGr}(\mathsf{Vect})$, and behaviours of spans are solutions of differential equations. Within the theory of continuous linear systems is of course the theory of linear RLC circuits; in fact, we shall see that there are a class of arrows in $\mathsf{Ifo}(\mathsf{Vect},\oplus)$ whose image under $\mathcal{D}$ are (models of) capacitors and whose image under $\mathcal{I}$ are inductors. The geometry associated to an expression in our algebra is, of course, the classical circuit picture. We end the section by indicating how the notion of an ifo system can be modified so as to obtain a more general theory of continuous input-output systems.

4.1 $\mathsf{Ifo}(\mathsf{Vect}, \oplus)$ and $\mathbf{SpGr}(\mathsf{Vect})$

Let Vect be the category of finite dimensional vector spaces over the reals $\mathbb{R}$. Recall that Vect is a linear category in that finite products equal finite sums (which is the direct sum $\oplus$). Given a graph $(p, v) : M \to S^2$ in Vect we define a *behaviour* of this graph to be a differentiable function $f : \mathbb{R} \to M$ such that

$$\frac{d}{dt}(p \cdot f) = (p \cdot f)' = v \cdot f.$$

(In order to define behaviours we have to go outside the world of graphs in Vect, which indicates that this is not really the appropriate place to model continuous systems. We will address this point at the end of this section.)

By a behaviour of a span of graphs we mean a behaviour of the head of the span. Of course, via composition with the legs, this induces behaviours of the boundaries.

For example, consider the identity arrow of an object X of $\mathbf{SpGr}(\mathsf{Vect})$. (Recall that X is a one vertex graph.) The head of the span is the graph $(!, !) : X \to 1^2$ and a behaviour of this span is just a differentiable function $f : \mathbb{R} \to X$.

A more interesting example is the span from X to X whose head is the graph $1 = (\pi_1, \pi_2) : X^2 \to X^2$, whose left leg is the first projection $\pi_1 : X^2 \to X$ and whose right leg is the second projection $\pi_2 : X^2 \to X$. A behaviour of this span is a function $(f, g) : \mathbb{R} \to X^2$ such that $f' = g$. Notice that the corresponding behaviours on left and right boundaries are f and g respectively. We call such an arrow a *differentiator*, since the behaviour on the right boundary is the derivative of the behaviour on the left.

Another example is obtained by interchanging the roles of position and velocity in the above example. More precisely, consider the span from X to X whose centre is the graph $twist_X : X^2 \to X^2$ and whose left and right legs are the first and second projections respectively. We call such a span an *integrator*, since an input (left boundary behaviour) is integrated, instantaneously appearing as an output (right boundary behaviour). Notice that the integrator is isomorphic to the opposite of the differentiator (that is, by swapping the left and right legs of the differentiator we obtain the integrator).

It is clear that the notion of behaviour is compositional, in the sense that if $\alpha : X \to Y$ and $\beta : Y \to Z$ are spans then to give a behaviour of $\beta \cdot \alpha$ is to give behaviours of α and β that agree on the common boundary Y. Of course, this fact can be expressed by describing a structure preserving homomorphism *behaviour* from $\mathbf{SpGr}(\mathsf{Vect})$ to a suitable semantic category.

Recall that an arrow in $\mathsf{Ifo}(\mathsf{Vect}, \oplus)$ from X to Y is a linear transformation of the form $X \oplus U \to U \oplus Y$. Noting that $U \oplus Y$ is in fact a product, we will write such arrows as $(\alpha, \beta) : XU \to UY$.

Let $\mathcal{D} : \mathsf{Ifo}(\mathsf{Vect}, \oplus) \to \mathbf{SpGr}(\mathsf{Vect})$ be the homomorphism described in Section 3. So $\mathcal{D}$ maps an arrow $(\alpha, \beta) : XU \to UY$ to the span from X to Y with head $(\pi_2, \alpha) : XU \to U^2$, left leg $\pi_1 : XU \to X$ and right leg $\beta : XU \to Y$.

Clearly, to give a behaviour of $\mathcal{D}(\alpha, \beta)$ is to give a pair of differentiable functions $(f, g) : \mathbb{R} \to XU$ such that $g' = \alpha(f, g)$. Using vector notation, a behaviour of the system (including the boundaries) is a triple $(x(t), u(t), y(t))$ such that

$$u' = \alpha \begin{pmatrix} x \\ u \end{pmatrix}$$
$$y = \beta \begin{pmatrix} x \\ u \end{pmatrix}.$$

Applying $\mathcal{D}$ to the delay δ_X yields the integrator described above. (So behaviours of an integrator are governed by a differential equation.) Of course, a behaviour of

$$\mathcal{D}(\mathsf{Fb}^{\mathbb{R}}_{I,I}(1_{\mathbb{R}})) \cong \mathsf{Tr}^{\mathbb{R}}_{I,I}(\mathcal{D}(\delta_{\mathbb{R}}))$$

is a scalar multiple $Ae^t : \mathbb{R} \to \mathbb{R}$ of the exponential function – it being a solution to the equation $x' = x$.

There is an involution **Graph** $\to$ **Graph** which is the functor mapping a graph to its dual (that is, it sends a graph $(s, t) : E \to V^2$ to the graph $(t, s) : E \to V^2$). This induces an involution $\mathsf{dual} : \mathbf{SpGr} \to \mathbf{SpGr}$ which is the identity on objects.

Let $\mathcal{I} : \mathsf{Ifo}(\mathsf{Vect}, \oplus) \to \mathbf{SpGr}(\mathsf{Vect})$ be the homomorphism $\mathsf{dual} \cdot \mathcal{D}$. So $\mathcal{I}$ maps an arrow $(\alpha, \beta) : XU \to UY$ to the span from X to Y with head $(\alpha, \pi_2) : XU \to U^2$, left leg $\pi_1 : XU \to X$ and right leg $\beta : XU \to Y$. Notice that the head of $\mathcal{I}(\alpha, \beta)$ is the dual graph of $\mathcal{D}(\alpha, \beta)$.

Clearly, to give a behaviour of $\mathcal{I}(\alpha, \beta)$ is to give a triple $(x(t), u(t), y(t))$ such that

$$u = \alpha \begin{pmatrix} x' \\ u' \end{pmatrix}$$
$$y = \beta \begin{pmatrix} x \\ u \end{pmatrix}.$$

Notice that we could view the above as an integral equation – that is, the integral of u from 0 to t is constrained to equal a linear function of x and u (plus a constant).

Applying $\mathcal{I}$ to the delay δ_X yields the differentiator described above. (So the behaviours of a differentiator are governed by an integral equation.) We invite the reader to (evaluate the expressions and) consider behaviours of $\mathcal{I}(\pi_1, \pi_2) \cdot \mathcal{D}(\pi_1, \pi_2)$ and $\mathcal{D}(\pi_1, \pi_2) \cdot \mathcal{I}(\pi_1, \pi_2)$ in relation to the fundamental theorem of calculus.

4.2 Towards an algebraic foundation for electrical circuit theory

We are now in a position to define the basic components needed to build RLC circuits. Let $X = \mathbb{R}^2$. We will be thinking of the first variable i of X as current and the second variable v as voltage. Resistors, capacitors and inductors will be defined to be certain spans $X \to X$. Of course, a behaviour of such a span induces a behaviour $(i(t), v(t), j(t), w(t)) \in (XX)^{\mathbb{R}}$ on the boundaries; it is these behaviours that we consider below.

Resistors For any $R \in \mathbb{R}$, let $\rho_R : X \to X$ be the transformation $(i, v) \mapsto (i, v - Ri)$. The resistor of resistance R is the span $\mathcal{D}(\mathsf{Inst}(\rho_R)) : X \to X$. Notice that this equals $\mathcal{I}(\mathsf{Inst}(\rho_R))$, and we denote this span also by ρ_R. A behaviour (i, v, j, w) of the resistor ρ_R satisfies, for all t, the condition $i = j$ and $w = v - Ri$. This latter equation can be written in the more familiar form $V = v - w = Ri$.

Capacitors For any $C \in \mathbb{R}$, let $\chi_C : X\mathbb{R} \to \mathbb{R}X$ be the transformation $(i, v, q) \mapsto (i, i, v - \frac{q}{C})$. The capacitor of capacitance C is the span $\mathcal{D}(\mathbb{R}, \chi_C) : X \to X$, which we denote by χ_C. The induced boundary behaviours (i, v, j, w) satisfy $i = j$ and $V' = (v - w)' = \frac{q'}{C} = \frac{i}{C}$.

Inductors For any $L \in \mathbb{R}$, let $\iota_L : X\mathbb{R} \to \mathbb{R}X$ be the transformation $(i, v, p) \mapsto (i, i, v - Lp)$. Notice that $\iota_L = \chi_{\frac{1}{L}}$. The inductor of inductance L is the span $\mathcal{I}(\mathbb{R}, \iota_L) : X \to X$ which we denote by ι_L. The induced boundary behaviours (i, v, j, w) satisfy $i = j$ and $V = (v - w) = Lp = Li'$.

Kirchhoff Wires The Kirchhoff diagonal wire is the span $W_{1,2}$ from X to X^2 which has as centre the graph $(!, !) : \mathbb{R}^3 \to 1^2$, as left leg the transformation $(i_1, i_2, v) \mapsto (i_1 + i_2, v)$ and as right leg the transformation $(i_1, i_2, v) \mapsto (i_1, v, i_2, v)$. It is important to note that this is not the diagonal of the discrete Cartesian structure of $\mathbf{SpGr}(\mathsf{Vect})$. Other Kirchhoff wires can be constructed from $W_{1,2}$ using the compact closed structure of $\mathbf{SpGr}(\mathsf{Vect})$.

RLC circuits The bicategory **RLC** of RLC circuits is the smallest self-dual compact closed sub-bicategory of $\mathbf{SpGr}(\mathsf{Vect})$ containing the above basic components. Rather than just an arrow in **RLC**, by an RLC circuit we sometimes mean an expression built out of resistors, inductors, capacitors and the Kirchhoff diagonal wire using only the self-dual compact closed structure of $\mathbf{SpGr}(\mathsf{Vect})$ (that is, composition, tensor product, the identity $1_X : X \to X$, the symmetry $s_{X,X} : XX \to XX$, the unit $\eta_X : 1 \to XX$ and the counit $\epsilon : XX \to 1$.) Of course, the pictures of these expressions are similar to the classical RLC circuit diagrams.

This theory permits a more subtle analysis than given by classical linear circuit theory by virtue of the fact that the internal state of a component is explicitly described. For example, the composite of two capacitors is not a capacitor as its internal state is $\mathbb{R}^2$. However, it has the same input-output behaviours as a capacitor (if started in appropriate initial states); in fact, the diagonal function $\Delta : \mathbb{R} \to \mathbb{R}^2$ provides a 2-cell from the capacitor $\chi_{\frac{C_1 C_2}{C_2 + C_1}}$ to the composite $\chi_{C_2} \cdot \chi_{C_1}$.

Though there is no reason to expect that an RLC circuit will be a deterministic system, we know of many interesting RLC circuits which do lie in the image of the homomorphism $\mathcal{D}$. For example, there are isomorphisms

$$\mathsf{Tr}^{X}_{I,I}(\iota_L \cdot \chi_C) \cong \mathsf{Tr}^{\mathbb{R}}_{I,I}(\delta_{\mathbb{R}} \cdot \rho_{\frac{1}{LC}} \cdot \delta_{\mathbb{R}}) \cong \mathcal{D}(\mathsf{Fb}^{\mathbb{R}}_{I,I}(\delta_{\mathbb{R}} \cdot \rho_{\frac{1}{LC}})).$$

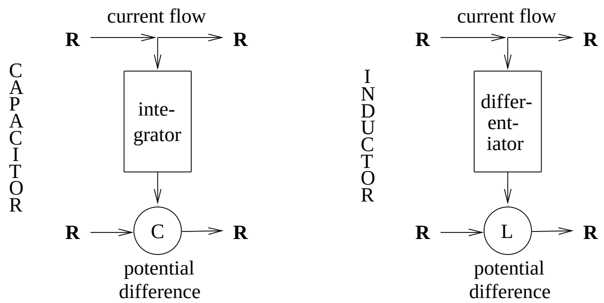
Given this fact, we can now find the behaviours of the RLC circuit $\mathsf{Tr}^{X}_{I,I}(\iota_L \cdot \chi_C)$ by using the theory of linear algebra to calculate eigenvalues of the linear transformation $\mathsf{Fb}^{\mathbb{R}}_{I,I}(\delta_{\mathbb{R}} \cdot \rho_{\frac{1}{LC}})$.

By going outside the algebra of RLC circuits it is possible to construct isomorphisms, such as the first one mentioned in the previous paragraph, by using the axioms of the discrete Cartesian structure ([13]) of **SpGr**(Vect). Some of these calculations involve the diagonal Δ, which (if it has any physical significance) should not be viewed as a wire (in the sense of Kirchhoff's law) but rather as a component which allows a current to affect several parts of a system.

It is possible to construct the isomorphism

$$\mathsf{Tr}^{X}_{I,I}(\iota_L \cdot \chi_C) \cong \mathsf{Tr}^{\mathbb{R}}_{I,I}(\delta_{\mathbb{R}} \cdot \rho_{\frac{1}{LC}} \cdot \delta_{\mathbb{R}})$$

by first decomposing the capacitor and the inductor into expressions involving diagonals and (images under $\mathcal{D}$ and $\mathcal{I}$ of) delays and instantaneous ifo systems, and then suitably rearranging the expression. The following pictures may illuminate these remarks.



The splittings of wires represent diagonal $\Delta : \mathbb{R} \to \mathbb{R}^2$ components and the functions which determine the potential difference are

$$C : \mathbb{R}^2 \to \mathbb{R} : (q, v) \mapsto v - \frac{q}{C}$$

and

$$L : \mathbb{R}^2 \to \mathbb{R} : (p, v) \mapsto v - Lp.$$

For the calculation of the above isomorphism, it is important to note that the differentiator $\mathcal{I}(\delta_{\mathbb{R}})$ is the opposite (or dual) span of the integrator $\mathcal{D}(\delta_{\mathbb{R}})$.

The reader may like to contrast this approach to electrical circuit theory with the one proposed in [44]. When modelling dynamical systems one could take as primary the underlying space of the system and the quantities (functions) varying over it (which leads to cohomology theories etc.) as is done in the previously cited paper. One can also take as primitive some notion of a process (which must entail concepts such as input, output and feedback) and, in this case, it is natural to ask what kind of algebra these processes form. It is this last question we are endeavouring to answer here. In the next subsection we provide a setting for modelling more general continuous systems (for example, non-linear circuits).

4.3 A parametrized theory of ifo systems and spans

The structures described earlier in this section should be understood in a more general context. Rather than considering a graph $(p, v) : M \to S^2$ as a model of a continuous system, we should consider the notion of a space S equipped with a generalized tangent bundle – namely, a map $M \to S^I$, where the object S^I is (the domain of) the tangent bundle of S (and I is the infinitesimal object of synthetic differential geometry [32], [37]).

In fact, we can develop much of the theory presented here parametrized by copointed endofunctors $\mathsf{position} : R \Rightarrow 1_{\mathbf{C}}$ which preserve products. The discrete case is given by $R(A) = A^2$ with $\mathsf{position} = \mathrm{proj}_1$ and the continuous case is given by $R(A) = A^I$ with $\mathsf{position} = \mathsf{ev}_0$.

The category of R-systems is the comma category $\mathbf{C}/R$. So an R-system is a pair of objects A and B together with a map $A \to R(B)$, and a morphism of R-systems from $\alpha : A \to R(B)$ to $\alpha' : A' \to R(B')$ is a pair of maps $f : A \to A'$ and $g : B \to B'$ such that $\alpha' \cdot f = R(g) \cdot \alpha$.

A deterministic R-system is a coalgebra for R – that is, a map $\alpha : A \to R(A)$ such that $\mathsf{position}_A \cdot \alpha = 1_A$. In the discrete case this is equivalent to an endomorphism and in the continuous case this is a vector field.

An ifo R-system with input X, internal state U and output Y is a map $(\alpha, \beta) : XU \to (RU)Y$ such that $\mathsf{position}_U \cdot \alpha = \mathrm{proj}_U : XU \to U$. (Note that, for the discrete case $R(A) = A^2$, an ifo R-system is equivalent to an ifo system as previously defined via the loops-suspension construction.) The collection of ifo R-systems naturally forms a monoidal bicategory.

We still obtain a monoidal homomorphism from the bicategory of ifo R-systems to spans of R-systems. There is a natural operation of R-feedback, but the notion of an R-delay (and the connection with the trace of **Span**) appears to be less straightforward.

The time object T (as discussed in Section 3.1) which parametrizes the behaviours of continuous systems is $(1_{\mathbb{R}}, \overline{1}) : \mathbb{R} \to \mathbb{R}^2 \cong \mathbb{R}^I$, where $\overline{1} : \mathbb{R} \to \mathbb{R}$ is the constant function at 1. It is straightforward to check that to give a behaviour of a continuous system X (as described in this section) is to give a morphism of continuous systems from T to X.

Of course, we were able to present the examples of this section in terms of graphs since, for all $X = \mathbb{R}^n$, we have $X^I \cong X^2$ (in an appropriate category of smooth maps). A relationship between the continuous and the discrete cases (which incorporates behaviours) could perhaps be understood by considering the passage between X^I, $X^{[0,1]}$ and X^2, where $[0, 1]$ is the interval.

The details of this general theory will be given elsewhere wherein we will also describe connections with the work on coalgebras done by Jacobs and Rutten [41].

5 Accounting and the continuity equation

Up to this point we have not seriously used the 2-dimensional structure of the bicategory **Span**(**Graph**). In order to obtain an algebra of systems such as accounts – which are essentially distributed dynamical systems with values recorded in the integers – we will use the 2-cells of **Span**(**Graph**).

A quantity varying over a space X is often viewed as a function from X into some ring. We will adapt this view to the present setting. The ring is replaced by a family of spans called standard accounts and, instead of considering a function from a space into a ring, we consider (roughly speaking) a 2-cell from a span into one of these standard accounts. Such a 2-cell is called an account and can be viewed as a system satisfying a discrete version of the continuity equation. Moreover, we have an algebra of such structures; in [27] we call an expression in this algebra a system of accounts with *partita doppia*, it being a formalization of double-entry accounting, first described in [38].

The construction of this algebra will take place in two stages. First we describe pre-accounts which have values only associated to their motions (=edges); these systems satisfy the condition that, for any motion, the total in-flow equals the total out-flow. Then we modify this construction in order to describe the standard accounts in which the value of a state (=vertex) of a system can be recorded. The section finishes with an indication of how the same basic construction can be used to model continuous systems satisfying the continuity equation ([10]).

Bicategorical preliminaries

Let us first recall a basic construction on bicategories. Suppose $f : A \to B$ and $g : C \to D$ are arrows in some bicategory $\mathcal{B}$. A *square* in $\mathcal{B}$ from f to g comprises two arrows $h : A \to C$ and $h' : B \to D$ together with a 2-cell of the following form:

$$\begin{tikzcd} A \arrow[r, "h"] \arrow[d, "f"'] \arrow[dr, phantom, "\Downarrow\alpha"] & C \arrow[d, "g"] \\ B \arrow[r, "i"'] & D \end{tikzcd}$$

We sometimes call f and g the *sides* of the square. Such squares naturally form the arrows of a bicategory $\mathbf{Sq}(\mathcal{B})$ in which composition is given by horizontally pasting squares (that is, pasting them side by side). If $\mathcal{B}$ admits a monoidal structure then so too does its bicategory of squares. Now consider the full and locally-full sub-bicategory $\mathbf{Ladj}(\mathcal{B})$ of $\mathbf{Sq}(\mathcal{B})$ whose objects $f : A \to B$ are left adjoints as arrows in $\mathcal{B}$. It is the case that if $\mathcal{B}$ is compact closed then so too is $\mathbf{Ladj}(\mathcal{B})$ ([30]).

Each morphism $f : G \to H$ in a category $\mathbf{G}$ determines a span f_* in $\mathbf{G}$ whose left leg is the identity and whose right leg is f. These spans f_* can be characterized as those arrows of $\mathbf{Span}(\mathbf{G})$ which are left adjoints. A square in $\mathbf{Span}(\mathbf{G})$ from f_* to g_* amounts to giving the following commutative diagram *in* $\mathbf{G}$ whose shape resembles a tent.

$$\begin{tikzcd}
 & H \arrow[dl] \arrow[dr] \arrow[dd] & \\
G \arrow[dd, "f"'] & & J \arrow[dd, "g"] \\
 & V \arrow[dl] \arrow[dr] & \\
H & & K
\end{tikzcd}$$

Via the above mentioned result we know that the bicategory $\mathbf{Ladj}(\mathbf{Span}(\mathbf{G}))$ of tents in $\mathbf{G}$ is compact closed. As pointed out by Aurelio Carboni, there is a direct way to see that $\mathbf{Ladj}(\mathbf{Span}(\mathbf{G}))$ is compact closed: it is (monoidally) biequivalent to the bicategory of spans in the functor category $\mathbf{G}^{\mathbf{2}}$, where $\mathbf{2}$ is the two-object category comprising the free living arrow $\bullet \longrightarrow \bullet$.

Pre-accounts

We now begin the description of pre-accounts. Let $\mathcal{C}$ be the chaotic category whose objects are finite sequences (ξ_i) of $+$'s and $-$'s. This category admits an obvious compact closed structure, the tensor product being juxtaposition of sequences. Let $C : \mathcal{C} \to \mathbf{Span}(\mathbf{Graph}(\mathsf{Set}))$ be the (lax) morphism of bicategories which maps (ξ_i) to the one-vertex graph with edge-set $\prod_i \mathbb{N}$ and the arrow $(\xi_i) \to (\zeta_j)$ to the span $C_{(\xi_i),(\zeta_j)}$ whose centre is the one-vertex graph with edge-set

$$\{((x_i),(y_j)) \in \prod_i \mathbb{N} \times \prod_j \mathbb{N} \quad | \quad \sum_i \xi_i x_i = \sum_j \zeta_j y_j\}$$

and whose legs are the evident projections. This data specifies C, since the structure 2-cells of the morphism are uniquely determined: that is, for all sequences (ξ_i), (ζ_j) and (ψ_k), there exist unique 2-cells

$$1_{\prod_i \mathbb{N}} \Rightarrow C_{(\xi_i),(\xi_i)} \quad \text{and} \quad C_{(\zeta_j),(\psi_k)} \cdot C_{(\xi_i),(\zeta_j)} \Rightarrow C_{(\xi_i),(\psi_k)}.$$

The spans $C_{(\xi_i),(\zeta_j)}$ are called *standard pre-accounts.* These spans have a close connection with the Kirchhoff wires described in the previous section on RLC circuits.

Suppose $S = (\xi_i)$ and $T = (\zeta_j)$ are strings of plus and minus signs and $f : X \to \prod_i \mathbb{N}$ and $g : Y \to \prod_j \mathbb{N}$ are graph morphisms. We define a *pre-account from* (S, f) *to* (T, g) to be a pair (R, α) where R is a span of graphs $R : X \to Y$

and α is a square in **Span**(**Graph**) of the form

$$\begin{tikzcd} X \arrow[r, "R"] \arrow[d, "f_*"'] \arrow[dr, phantom, "\Downarrow\alpha"] & Y \arrow[d, "g_*"] \\ \prod_i \mathbb{N} \arrow[r, "C_{S,T}"'] & \prod_j \mathbb{N} \end{tikzcd}$$

We think of this structure as recording the flow of quantities across the boundaries of a system R for which the total in-flow of each transition equals the total out-flow. Note that as the centre of the span $C_{S,T}$ is a one-vertex graph, the pre-account does not associate any values to the vertices of R.

Preaccounts do not form a sub-bicategory of **Ladj**(**Span**(**Graph**)); this is because the composite of two standard pre-accounts is not necessarily another preaccount. However, the structure 2-cells of the morphism C can be used to define a bicategory of pre-accounts. For example, if $(R, \alpha) : (S, f) \to (T, g)$ and $(P, \beta) : (T, g) \to (U, h)$ are pre-accounts then their composite is defined to be $(P \cdot R, \theta)$ where θ is the following square (the unlabelled 2-cell is part of the morphism's lax structure).

$$\begin{tikzcd} X \arrow[r, "R"] \arrow[d, "f_*"'] \arrow[dr, phantom, "\Downarrow\alpha"] & Y \arrow[r, "P"] \arrow[d, "g_*"] \arrow[dr, phantom, "\Downarrow\beta"] & Z \arrow[d, "h_*"] \\ \prod_i \mathbb{N} \arrow[r, "C_{S,T}"'] \arrow[rr, bend right, "C_{S,U}"', ""{name=B}] & \prod_j \mathbb{N} \arrow[r, "C_{T,U}"'] \arrow[d, phantom, "\Downarrow"] & \prod_k \mathbb{N} \\ & {} & \end{tikzcd}$$

The morphism C also has a lax structure with respect to tensor products: for example, for any sequences S, T, U and V, there exists a unique 2-cell $C_{S,T} \otimes C_{U,V} \Rightarrow C_{S\otimes U, T\otimes V}$. The reader can find the details of this structure in [27]; let us say here that this structure is enough to make the bicategory of pre-accounts compact closed. An expression in this compact closed bicategory can be thought of as a system whose components can make transactions across the boundaries on which they are joined, but where each component's net transaction is zero. Perhaps one could use this algebra to give a simplistic model of a barter system: transactions cannot incur debit nor credit and the accumulation of value is not possible.

Accounts

We will now modify the above morphism $C : \mathcal{C} \to$ **Span**(**Graph**) in order to obtain the model of accounting described in [27]; essentially this is done by introducing internal state into the standard pre-accounts.

The *standard account from + to +* is the span of graphs from $\mathbb{N}$ to $\mathbb{N}$ whose centre has as vertex-set the set of integers, and wherein an edge $\rho : r \to s$ is a pair (x, y) of natural numbers such that $s - r = x - y$. The left leg of the span maps (x, y) to x and the right leg maps (x, y) to y. We denote this span by $A_{+,+}$. We think of the edge $(x, y) : r \to s$ as a motion of the system which on the left boundary corresponds to an input (or in-flow) of x and on the right boundary corresponds to an output (or out-flow) of y.

If $S = (\xi_i)$ and $T = (\zeta_j)$ are strings of $+$'s and $-$'s then the standard account $A_{S,T}$ from S to T is the composite $C_{+,T} \cdot A_{+,+} \cdot C_{S,+}$ of standard pre-accounts with the standard account from $\mathbb{N}$ to $\mathbb{N}$. An edge $\rho : r \to s$ in (the centre of) this standard account is a pair of tuples of natural numbers (x_i) and (y_j) such that

$$s - r = \sum_i \xi_i x_i - \sum_j \zeta_j y_j,$$

this equation being a discrete version of the continuity equation. Such an edge ρ should be thought of as having in-flows (resp. out-flows) on those boundaries k for which ξ_k is positive (resp. negative) or ζ_k is negative (resp. positive). Below we picture some transitions in a standard account (the arrowheads on the boundary wires are used to indicate that the domain is $+-$ and that the codomain is $+$).

$$5 \xrightarrow{(3,2)/6} 0 \xrightarrow{(12,0)/2} 10 \xrightarrow{(5,100)/5} -90$$

Accounts are defined in a similar way to pre-accounts. Suppose $S = (\xi_i)$ and $T = (\zeta_j)$ are strings of plus and minus signs and $f : X \to \prod_i \mathbb{N}$ and $g : Y \to \prod_j \mathbb{N}$ are graph morphisms. An *account from* (S, f) *to* (T, g) comprises a span of graphs $R : X \to Y$ and a square of the form

$$\begin{tikzcd} X \arrow[r, "R"] \arrow[d, "f_*"'] \arrow[dr, phantom, "\Downarrow\alpha"] & Y \arrow[d, "g_*"] \\ \prod_i \mathbb{N} \arrow[r, "A_{S,T}"'] & \prod_j \mathbb{N}. \end{tikzcd}$$

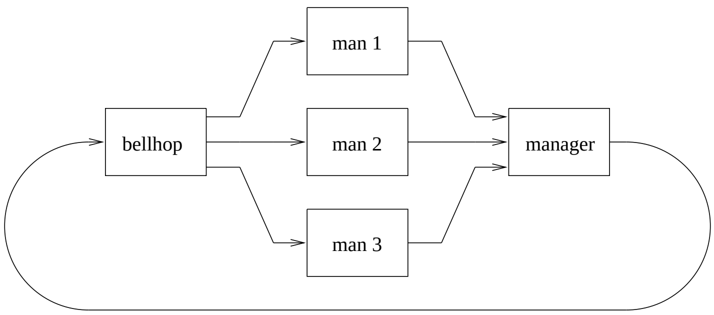
Notice that an account has an integer associated to each vertex (of R, in the above case) as well as a tuple of natural numbers associated to each edge. Quantities can be stored in accounts, unlike in pre-accounts.

Standard accounts also define a (lax) morphism $A : \mathcal{C} \to \mathbf{Span}(\mathbf{Graph})$ of compact closed bicategories. For example, the standard account $A_{+,+}$ naturally admits the structure of a monad in $\mathbf{Span}(\mathbf{Graph})$: the unit $1_{\mathbb{N}} \Rightarrow A_{+,+}$ is the 2-cell which picks out the vertex 0 of (the centre of) $A_{+,+}$ and the multiplication 2-cell $A_{+,+} \cdot A_{+,+} \Rightarrow A_{+,+}$ is defined (on vertices) to be the addition of integers. Of course, the monad axioms in this case amount to the monoid axioms for addition on the integers.

In the same way that the lax structure of the morphism C was used to define operations on preaccounts, the structure 2-cells of the morphism A are needed to define the operations on accounts. Again, the reader can find the details in

[27] (as well as the connection with traditional accounting), but let us say here that these structure 2-cells should be called *total* or *add* and their coherence (in the sense that they are part of the data for a morphism of bicategories) trivially implies that for a closed system of accounts – that is, an expression of accounts whose domain and codomain are of the form $(\emptyset, ! : X \to I)$ – there is an invariant of a behaviour called the *total value*. In order to see this, note that the standard account $A_{\emptyset,\emptyset}$ from the empty string to the empty string has as its centre the discrete reflexive graph whose vertex-set is the set of integers, and so any 2-cell into $A_{\emptyset,\emptyset}$ must map every connected component of the domain to an integer.

For example, consider the simplistic *albergo problem*. Three men decide to take a room in a hotel. The manager of the hotel charges them \$30 for a room with three beds. After paying the manager \$10 each, they retire to their room. The manager then realizes that he has overcharged the men as the room only costs \$25, so he gives \$5 to the bellhop to refund the men. The bellhop decides it would be easier to divide \$5 among three men if he first pockets \$2, which he does, then giving \$1 to each man. The supposed brain teaser is: each man has paid \$9 and \$2 has gone to the bellhop, so where has the extra \$1 (from the orginal \$30) gone? Below we depict the accounting system corresponding to the albergo problem.



Each component is a standard account; for example, the manager is modelled by the standard account from $+++$ to $+$. Suppose that the system starts in state $(0,0,0,0,0)$. The behaviour of this system corresponding to the events in the above problem is

$$(0,0,0,0,0), (30,-10,-10,-10,0), (25,-10,-10,-10,5), (25,-9,-9,-9,2)$$

and, of course, the total is conserved. (So the correct calculation to be performed at the end is $25+2 = 9+9+9$, not $9+9+9+2 = 30$.)

Given that each of the three men in the albergo problem started in state 0, we might imagine that they were using credit cards to pay. Credit cards usually have a limit, so they should not be modelled by standard accounts. We can model a credit card with credit limit $n \in \mathbb{N}$ as the account from $(+, 1_{\mathbb{N}})$ to $(+, 1_{\mathbb{N}})$ defined by the full subgraph B_n of $A_{+,+}$ comprising all the vertices greater than or equal to $-n$.

As any expresssion of standard accounts admits a 2-cell into the standard account with the same domain and codomain as the expression, standard accounts can be used to build more complicated accounts. For example, the three men could together be viewed as an account from $(+++, 1_{\mathbb{N}^3})$ to $(+++, 1_{\mathbb{N}^3})$. In examples like this, more structure may be involved as *joint* accounts are often used.

More generally, suppose a complex system is modelled by a complicated expression of standard accounts in which certain subexpressions correspond to parts of the system (the system may be a business, and the subsytems may be different business divisions). At some point during the life cycle of the system, we may want to total the value of a particular part of the system. This could be done if the sub-expression corresponding to this part is viewed as an account itself. It is our contention that this algebra of accounts provides a framework (for future work) which will clarify the hierarchical structures and types of abstraction processes used in the accounting of complex systems.

Also, in the above examples, we have only described accounts with sides that are identities (in other words, the squares are just 2-cells). To model systems where the in-flow and out-flow of accounts are more than just numbers (for example, they may be more concrete representatives of items), or where transactions are governed by certain protocols, we need to consider genuine squares.

'Fluid flows'

We will now see that structures similar to accounts can be defined in the world of continuous systems. So for the rest of this section we will be working in the context of R-systems (as described at the end of Section 4), where R is the copointed endofunctor on a category of smooth spaces (admitting the operations of synthetic differential geometry) which maps a space A to its tangent space A^I.

To indicate what can be done, we will describe the continuous version of the standard account $A_{+,+}$. Its domain and codomain is the single state system $! : \mathbb{R} \to 1^I$, and its centre is the system $\alpha = (p, v) : \mathbb{R}^3 \to \mathbb{R}^2 \cong \mathbb{R}^I$, where $p(x, y, z) = y$ and $v(x, y, z) = x - z$. Its left leg is $\mathrm{proj}_1 : \mathbb{R}^3 \to \mathbb{R}$ and its right leg is $\mathrm{proj}_3 : \mathbb{R}^3 \to \mathbb{R}$. So a behaviour of this system is a triple $(x(t), y(t), z(t))$ such that $y' = x - z$. We denote this span by E. We saw earlier that the addition of integers equipped the standard account $A_{+,+}$ with a monad structure. In a similar way, the addition of reals equips the span E with a monad structure.

To see how E can act as the codomain for a system governed by the continuity equation, let us consider a model of a one dimensional 'fluid flow'. Let $[a, b]$ be an interval and suppose $\beta : M \to (\mathbb{R}^{[a,b]} \times \mathbb{R}^{[a,b]})^I$ is a system. So the states of this system are pairs $\rho, u : [a, b] \to \mathbb{R}$ where we are thinking of ρ as a density function and u as a velocity field. Let us accordingly write the function β as (ρ, u, ρ', u') and let us also write $\beta(m)$ as $(\rho_m, u_m, \rho'_m, u'_m)$. Now consider the span from $\mathbb{R}$ to $\mathbb{R}$ whose centre is the above system and whose left leg is $l : M \to \mathbb{R} : m \mapsto \rho_m(a)u_m(a)$ and whose right leg is $r : M \to \mathbb{R} : m \mapsto \rho_m(b)u_m(b)$. Call this span S.

A 2-cell of spans from S to E comprises a pair of maps $f = (f_1, f_2, f_3) : M \to R^3$ and $g : \mathbb{R}^{[a,b]} \times \mathbb{R}^{[a,b]} \to \mathbb{R}$ such that $\beta \cdot f = g^I \cdot \alpha$, $f_1 = l$ and $f_3 = r$. Call such a 2-cell *total* if g is the function $g : (\rho, u) \mapsto \int_a^b \rho\, dx$. It is straightforward to check that if S admits a total 2-cell then any behaviour $(\rho(t), u(t))$ of S will satisfy the continuity equation

$$\frac{d}{dt}(\int_a^b \rho\, dx) = \rho(a)u(a) - \rho(b)u(b).$$

When generalizing this to higher-dimensional fluid flows, it would be appropriate to use 'total' squares instead of just 2-cells, the sides of the squares being functions which integrate $\rho \boldsymbol{u} \cdot \boldsymbol{n}$ over the boundaries.

6 Conclusions

The pair of bicategories $\mathsf{Ifo}(\mathbf{C}, \otimes)$ and $\mathbf{Span}(\mathbf{Graph})$ yield a coherent theory of systems which spans non-deterministic finite state automata, algorithms, concurrent distributed systems, asynchronous circuits and continuous linear systems. Systems form an algebra and hence the theory is compositional. Associated with the algebra is a geometry which accurately records the spatial distribution of the system.

References

1. Abramsky S, Interaction categories (extended abstract), in: Theory and Formal Methods Workshop, Springer Verlag, 1993.
2. Abramsky S, Semantics of Interaction, http://www.dcs.ed.ac.uk/home/samson/
3. Arnold A, *Finite transition systems*, Prentice Hall, 1994.
4. Bénabou J, Introduction to bicategories, in: Reports of the Midwest Category Seminar, Lecture Notes in Mathematics 47, pages 1–77, Springer-Verlag, 1967.
5. Bloom S and Ésik Z, *Iteration Theories: The Equational Logic of Iterative Processes*, EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1993.
6. Bloom S, Sabadini N, Walters RFC, Matrices, machines and behaviors, in: Applied Categorical Structures, 4: 343-360, 1996.
7. Blum L, Shub M, Smale S, On a theory of computation and complexity over the real numbers, NP-completeness, recursive functions and universal machines, in: Bull. Amer. Math. Soc., 21:1-46, 1989.
8. Brown C, Jeffrey A, Allegories of circuits, in: Proc. Logical Foundations of Computer Science, St. Petersburg, 1994.
9. Brzozowski JA, Seger C-J.H., *Asynchronous circuits*, Monographs in Computer Science, Springer Verlag, 1995.
10. Chorin AJ, Marsden JE, *A Mathematical Introduction to Fluid Mechanics*, Springer-Verlag, New York, 1990.
11. Carboni A, Kelly GM, Verity D, Wood R, A 2-categorical approach to change of base and geometric morphisms II, in: Theory and Applications of Categories, Vol. 4. No. 5, 1998, pp. 82-136.

12. Carboni A, Kelly GM, Wood R, A 2-categorical approach to change of base and geometric morphisms I, in: Cahiers de Topologie et Géométrie Différentielle Catégoriques, 32:47-95, 1991.
13. Carboni A, Walters RFC, Cartesian Bicategories I, in: Journal of Pure and Applied Algebra, 49, pages 11-32, 1987.
14. Day B and Street R, Monoidal bicategories and Hopf algebroids, in: *Adv. in Math.*, Vol 129, No 1 (1997) 99-157.
15. Freyd P and Yetter D, Braided compact closed categories with applications to low-dimensional topology, in: Advances in Math., 77, 156-182, 1989.
16. Gates R, On the Generic solution to $P(X) \cong X$ in Distributive Categories, in: Journal of Pure and Applied Algebra, 125 (1998), no. 1-3, 191–212.
17. Gates R, Katis P, Walters RFC, A program for computing with the Cartesian bicategory Span(Graph), School of Mathematics and Statistics, University of Sydney, 1998.
18. Gates R, Katis P, Sabadini N, Walters RFC, Automata with boundary, In preparation.
19. Holzmann GJ, *Design and Validation of Computer Protocols*, Prentice Hall, 1991.
20. Jones G, Sheeran M, Circuit design in Ruby, in: Formal methods for VLSI Design (North-Holland, Amsterdam, 1990) 13-70.
21. Joyal A, Street R, Braided tensor categories, in: Advances in Math., 102, 20-78, 1993.
22. Joyal A, Street R, Verity D, Traced monoidal categories, in: Math. Proc. Camb. Phil. Soc., 119, 447-468, 1996.
23. Katis P, Categories and bicategories of processes, PhD Thesis, University of Sydney, 1996.
24. Katis P, Sabadini N, Walters RFC, Bicategories of processes, in: Journal of Pure and Applied Algebra, 115, 141-178, 1997.
25. Katis P, Sabadini N, Walters RFC, Span(Graph): an algebra of transition systems, in: Proceedings AMAST '97, SLNCS 1349, 322-336, 1997.
26. Katis P, Sabadini N, Walters RFC, Representing P/T nets in Span(Graph), in: Proceedings AMAST '97, SLNCS 1349, 307-321, 1997.
27. Katis P, Sabadini N, Walters RFC, On Partita Doppia, to appear in: Theory and Applications of Categories.
28. Katis P, Sabadini N, Walters RFC, Weld H, Categories of pulse circuits with data types, In preparation.
29. Katis P, Sabadini N, Walters RFC, Weld H, An algebra of asynchronous binary circuits, In preparation.
30. Katis P, Walters RFC, The compact closed bicategory of left adjoints, available at http://cat.maths.usyd.edu.au/~giuliok
31. Kelly GM, Laplaza LM, Coherence for compact closed categories, in: Journal of Pure and Applied Algebra, 19:193-213, 1980.
32. Lawvere FW, Categorical Dynamics, (1967 Chicago Lectures) *Topos-Theoretic Methods in Geometry*, Aahrus 1979.
33. Lawvere FW, State Categories and Response Functors. Preprint 1986.
34. Lawvere FW, State Categories, Closed Categories, and the Existence of Semicontinuous Entropy Functions, IMA Research Report 86, University of Minnesota (1986).
35. Lawvere FW, Tools for the advancement of objective logic: closed categories and toposes, in: The Logical Foundations of Cognition (Oxford University Press), ed. Macnamara J, Reyes G E, 1993.

36. Lawvere FW, Schanuel SH, *Conceptual Mathematics: A First Introduction to Categories*, Cambridge University Press, 1997.
37. Moerdijk I, Reyes GE, *Models for Smooth Infinitesimal Analysis*, Springer-Verlag, New York, 1991.
38. Fra Luca Pacioli, *Summa de arithmetica geometrie proportzioni e proportionalita*, Venezia, 1494.
39. Penrose R, Applications of negative dimensional torsors, in: Combinatorial Mathematics and its applications, (D. J. A. Welsh, Ed.) pp. 221-244, Academic Press, New York, 1971.
40. Rugh WJ, *Linear System Theory*, Second Edition, Prentice Hall , 1996.
41. Rutten JJMM, Universal coalgebra: a theory of systems, Report CS-R9652, CWI 1996.
42. Sabadini N, Vigna S, Walters RFC, A note on recursive functions, in: Mathematical Structures in Computer Science, 6, 127-139, 1996.
43. Shields M.W., *An introduction to Automata Theory* (Blackwell Scientific Publications, Oxford, 1987).
44. Smale S, On the mathematical foundations of electrical circuit theory, in: J. Differential Geometry, 7, 193-210, 1972.
45. Sutherland I, Micropipelines, in: Communications of the ACM, 32(6), 1989.
46. Valmari A, On-the-fly verification with stubborn sets, in: Proceedings CAV93, SLCS 697, 397-408, Springer Verlag, 1993.
47. Valmari A, Error detection by reduced reachability graph generation, Preprint, 1988.
48. Vigna S, On the Relations between Distributive Computability and the BSS Model, in: TCS 162(1): 5-21 (1996)
49. Walters RFC, Data types in a distributive category, Bull. Austr. Math. Soc., 40:79–82, 1989.
50. Walters RFC, *Categories and Computer Science*, Carslaw Publications 1991, Cambridge University Press 1992.
51. Walters RFC, An imperative language based on distributive categories, in: Mathematical Structures in Computer Science, 2:249–256, 1992.
52. Weld H, *On categories of asynchronous circuits*, PhD Thesis, University of Sydney, 1998.