Representing Place/Transition Nets in Span(Graph)

Piergiulio Katis,
School of Mathematics and Statistics,
University of Sydney, N.S.W. 2006, Australia
e_mail: katis_p@maths.usyd.edu.au

N. Sabadini,
Dipartimento di Scienze dell'Informazione,
Università di Milano, Via Comelico 39/41, I-20135 Milano MI, Italy
FAX: +39-2-55006276; e_mail: sabadini@dsi.unimi.it

R.F.C. Walters.
School of Mathematics and Statistics,
University of Sydney, N.S.W. 2006, Australia
e_mail: walters_b@maths.usyd.edu.au

21st May 1997

Abstract

The compact closed bicategory **Span** of spans of reflexive graphs is described and it is interpreted as an algebra for constructing specifications of concurrent systems. We describe a procedure for associating to any Place/Transition system Ω an expression Ψ_Ω in the algebra **Span**. The value of this expression is a system whose behaviours are the same as those of the P/T system. Furthermore, along the lines of Penrose's string diagrams, a geometry is associated to the expression Ψ_Ω which is essentially the same geometry as that usually associated to the net underlying Ω.

1 Introduction

This article is part of a project to study bicategories of processes ([KSW97a], [BSW96],[K96],[KSW94],[SWW93]) in order to develop the algebraic foundations of concurrency. Much of this work has focused on deterministic systems and is closely related to the notion of a distributive automaton ([SW93]); in fact, a construction of elemetary nets using functional pulse circuits and synchronizers was given in [MSSW95]. This paper concerns itself only with non-deterministic systems. Though closely related, the resulting algebraic structures are not the same as in the deterministic case.

In the following section we describe an algebra **Span** for constructing specifications of communicating systems; the structure involved is that of a compact closed category ([KL80]). The algebra **Span** is actually a bicategory ([B67]) rather than a category. For the purposes of this paper, however, we will ignore the 2-cell structure; in particular, many of the equalities we state are in fact isomorphisms. A more detailed study of **Span** is given in [KSW97b]. Along the lines of Penrose's tensor diagrams ([P71]), we will associate a geometry to each expression of this algebra; such a geometry is to be interpreted as a depiction of the spatial distribution of a communicating system. For a detailed study of the geometry of tensor algebras, the reader is referred to [JS91] and [JSV96] – those familiar with these papers will notice that our diagrams are to be read from left to right rather than up the page.

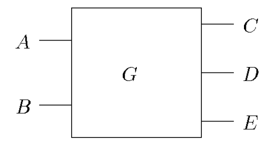
In Section 3 the basic definitions concerning Place/Transition systems are recalled, after which two classes of spans are described: one that corresponds to the places of a P/T system and another that corresponds to the transitions. Given a P/T system Ω we define an expression Ψ_Ω in **Span** constructed from these two classes of spans such that the (states and) behaviours of Ψ_Ω are precisely those of Ω; furthermore, the geometry of the expression Ψ_Ω is equivalent to the geometry usually associated to the net underlying the P/T system Ω. Of course, not every expression constructed from these two classes of spans can be interpreted as a P/T system; we do, however, identify a class of expressions which do admit such interpretations. (Note that this use of algebra is different from the way monoidal categories were used in [MM90] to model Petri nets. In the aforementioned paper a net was defined to be a certain monoidal category; this paper uses a compact closed category to construct general nets.)

For an introduction to category theory, the reader is referred to [M70] or [W92a].

2 The algebra of spans

Let **RGraph** be the category of reflexive graphs. The specified edge at each vertex v of a graph (that is determined by the reflexivity structure) is referred to as the identity edge at v.

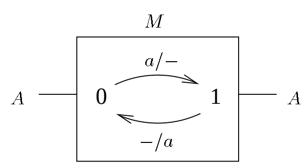
We now begin describing the algebra **Span**. Its objects are one-vertex graphs – the reader may think of a one-vertex graph as an alphabet with a null symbol (the null symbol corresponding to the identity edge). Given two one-vertex graphs A and B, an arrow in **Span** from A to B is a diagram in **RGraph** of the form $A \xleftarrow{l} G \xrightarrow{r} B.$ This is called a *span* of graphs from A to B: the graph G is called its *head* and l and r are respectively referred to as its left and right *legs*; we denote this arrow of **Span** by $G : A \to B$. Often a span will be given in the form where its domain and codomain are products of one-vertex graphs; for example, consider a span of the form $G : A \times B \to C \times D \times E$. We draw such a span as follows.



The objects A and B are called the *left boundaries* and C, D and E the *right boundaries* of G. Note that the span $G : A \times B \to C \times D \times E$ is just a labelling of G by $A \times B \times C \times D \times E$; in other words, it is a labelled transition system ([A94]) whose labelling is distributed among five alphabets. A more detailed comparison with existing work on transition systems is given in [KSW97b].

A *behaviour* of a span $G : A \to B$ is a finite path in the graph G. Notice that for each behaviour β of G there are corresponding behaviours (that is, sequences of letters) $l(\beta)$ and $r(\beta)$ of the boundaries A and B.

The first simple example of an 'agent' considered in [M89] is that of a message passer which can hold one datum. We would specify such a system by the span $M : A \to A$ pictured below. (Here, A is the one-vertex graph which has only one non-identity edge a.)



Notice that in depicting graphs we do not draw the identity edge at each vertex. The left and right legs are defined by the labelling of the edges. The symbol '−' denotes a labelling by the identity edge. The vertices (or states) 0 and 1 respectively correspond to the system having no data and having a datum. The edges $a/-$ and $-/a$ respectively correspond to the system receiving and sending a datum.

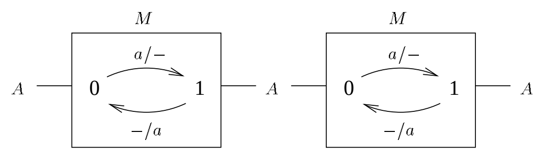
The first operation we define is *composition* (or series) of spans. Given two spans $G : A \to B$ and $H : B \to C$ their composite $G;H : A \to C$ is defined as follows. First form a diagram

$$\begin{tikzcd} & & K \arrow[dl, "\pi_0"'] \arrow[dr, "\pi_1"] & & \\ & G \arrow[dl, "l_G"'] \arrow[dr, "r_G"] & & H \arrow[dl, "l_H"'] \arrow[dr, "r_H"] & \\ A & & B & & C \end{tikzcd}$$

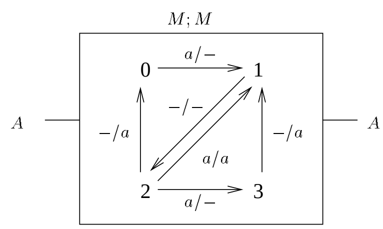
in **RGraph** having a pullback as its central square. The composite $G;H$ is $\begin{tikzcd} & K \arrow[dl, "l_G \cdot \pi_0"'] \arrow[dr, "r_H \cdot \pi_1"] & \\ A & & C. \end{tikzcd}$ Explicitly, a vertex of $G;H$ is a pair (v, w) of vertices

– a vertex v of G and vertex w of H. An edge of $G;H$ is a pair of edges (g,h) – g is an edge of G and h is an edge of H – which agree on their common boundary B. In this case, it would be reasonable to call B an interface between G and H. Later in this section we will introduce a more general notion of interface.

Composites are drawn in series. For example, the expression $M;M$ (that is, two message passers in series) is drawn as follows.



The value of the expression $M;M$ is the span depicted below.



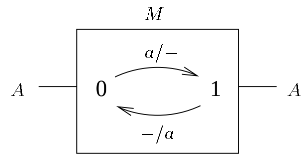
The states $0 = (0,0)$, $1 = (1,0)$, $2 = (0,1)$ and $3 = (1,1)$ respectively correspond to the system having no data, one datum in its first component, one datum in its second componenet and, lastly, two data items – one in each component. One interpretation of this system is that it specifies a bounded buffer of capacity two (that is, a message passer whose memory is a queue of length two).

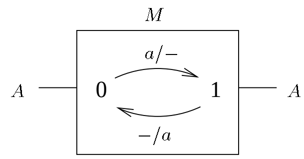
It is perhaps worthwhile to point out that in general there is a distinction between an expression in an algebra and its value; in particular, many expressions may yield the same value. Most often, of course, we will want to record both the expression and its value.

The other operation we are going to define is the *tensor product* (or parallel). If A and B are objects of **Span** then their tensor product $A \otimes B$ is defined to be the product $A \times B$ of graphs in **RGraph** (this is, of course, also a one vertex graph). If $G : A \to B$ and $H : C \to D$ are spans, their tensor product $G \otimes H : A \otimes C \to B \otimes D$ is the span

$$\begin{tikzcd} & G \times H \arrow[dl, "l_G \times l_H"'] \arrow[dr, "r_G \times r_H"] & \\ A \times C & & B \times D. \end{tikzcd}$$

Tensor products of spans are drawn in parallel. For example, the expression $M \otimes M : A \otimes A \to A \otimes A$ is drawn as follows.



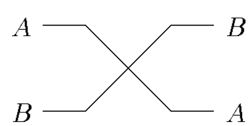


Let I be the terminal object of **RGraph**; that is, it is the one-vertex graph whose only edge is the identity edge. This object satisfies the property that $A \otimes I = A = I \otimes A$ and it plays the role of the 'null' boundary; note that to give a span $G : I \to I$ is just to give a graph G (since there is no information involved in giving the unique graph morphism $!_G : G \to I$). If a boundary of a span is I we usually omit it from the diagram.

We now introduce some constants; that is, special classes of spans. These constants will allow us to construct interfaces between more general spans.

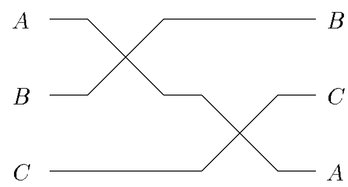
For any object A, the span
$$\begin{tikzcd} & A \arrow[dl, "1_A"'] \arrow[dr, "1_A"] & \\ A & & A \end{tikzcd}$$
is denoted $1_A : A \to A$ or merely A. It is the *identity* for composition in the sense that for any $G : A \to B$ or any $H : C \to A$, we have $A; G = G$ and $H; A = H$. This span is drawn as a line from A to A.

The second type of constant we are going to define is a *permutation*. For any pair of objects A and B, there is a graph morphism $\mathrm{tw} : A \times B \to B \times A : (a, b) \mapsto (b, a)$. The permutation $\pi_{A,B} : A \otimes B \to B \otimes A$ is defined to be the span
$$\begin{tikzcd} & A \times B \arrow[dl, "1_{A \times B}"'] \arrow[dr, "\mathrm{tw}"] & \\ A \times B & & B \times A. \end{tikzcd}$$
It is drawn as follows.



The crossing of the lines is not supposed to indicate that the boundaries A and B are in contact.

Given a bijective function $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ and objects $A_1, \ldots, A_n$, we can use spans of the form $\pi_{A,B}$ to construct a more general permutation $\pi : A_1 \otimes \ldots \otimes A_n \to A_{\pi(1)} \otimes \ldots \otimes A_{\pi(n)}$ in **Span**. For example, the following diagram depicts such a permutation.



The final two types of constants are the *unit* and *counit* for the self-dual compact closed structure. For any object A, let the unit $\eta_A : I \to A \otimes A$ be the span

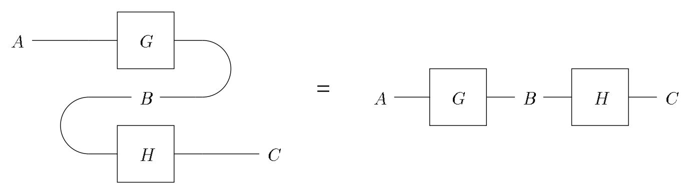
$$\begin{tikzcd} & A \arrow[dl, "!_A"'] \arrow[dr, "\Delta_A"] & \\ I & & A \times A, \end{tikzcd}$$

where $!_A$ is the unique map to the terminal graph and $\Delta_A : A \to A \times A : a \mapsto (a, a)$ is the diagonal; also, let the counit $\epsilon_A : A \otimes A \to I$ be the span

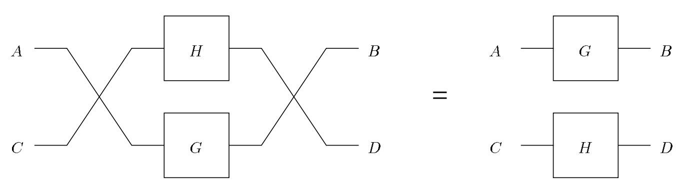
$$\begin{tikzcd} & A \arrow[dl, "\Delta_A"'] \arrow[dr, "!_A"] & \\ A \times A & & I. \end{tikzcd}$$

The spans η_A and ϵ_A are respectively drawn as follows.

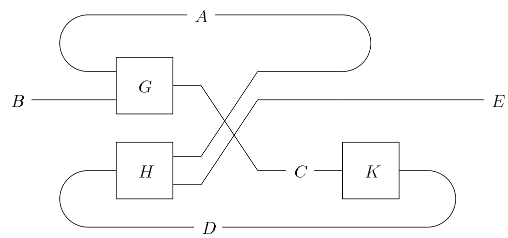
In describing the compact closed structure of **Span** we have defined two operations and four types of constants. We will not give the axioms that these data must satisfy. (For this, the reader can see [KL80].) A remarkable consequence of these axioms is that any two expressions are equal if their geometries are equivalent. For example, for any $G : A \to B$ and $H : B \to C$, the equality $(A \otimes \eta_B); (G \otimes 1_B \otimes H); (\epsilon_B \otimes C) = G; H$ can be derived. In terms of diagrams we have the following equality. (The reader can check that they are the same in **Span**.)



Another example of an equation that can be derived from the axioms is $\pi_{A,C}; H \otimes G; \pi_{D,B} = G \otimes H$. The corresponding equality of diagrams is depicted below.



A *concurrent system* comprising the spans $M_1, \ldots, M_n$ is defined to be an expression constructed from $M_1, \ldots, M_n$ and the constants described above using the operations composition and tensor product. Roughly speaking, the *interfaces* of a concurrent system are those parts of the system constructed from only the constants. We will not make this notion precise but give an example. In the concurrent system comprising G, H and K depicted below



B and E are external interfaces (in fact, they are boundaries of the value of the concurrent system) and A, C and D are internal interfaces. A vertex of this system is a triple $(u, v, w) \in G \times H \times K$ of vertices; and an edge is a triple $(g, h, k) \in G \times H \times K$ of edges such that i) k and h agree on the interface D, ii) h and g agree on A and iii) g and k agree on C.

The reader may have noticed that our notion of interface is limited in that a boundary of a span can be in contact with at most one other boundary. In order to express a more general type of communication – in which a boundary can be in contact with several other boundaries – we need to consider the discrete Cartesian structure ([CW87]) of **Span** (from which the compact closed structure can be recaptured). This is described in [KSW97b].

3 Place/Transition systems and their representations

The following definition of a net is taken from [T86].

Definition 1 A *net* N is a quadruple (S, T, F, G) where

- S and T are non-empty finite sets and
- $F \subseteq S \times T$ and $G \subseteq T \times S$ are subsets

such that

- for all $s \in S$ there exists $t \in T$ such that $(s, t) \in F$ or $(t, s) \in G$ and
- for all $t \in T$ there exists $s \in S$ such that $(s, t) \in F$ or $(t, s) \in G$.

The set S is called the set of *places* of N (sometimes also called the set of states) and T, the set of *transitions* of N.

We adopt the following notation: $\aleph$ is the (ordered) set of natural numbers; $\aleph^{>0}$ is the set of positive natural numbers; if X is an ordered set then $X \cup \infty$ is formed by formally adjoining a top element to X; for any natural number n let $[n] = \{0, \ldots, n\}$; and we define $[\infty]$ to be $\aleph$.

The following definition is taken from [R86]; the reader will notice that we have not included the specification of an initial marking in our defintion of a P/T system.

Definition 2 A *Place/Transition* system Ω (often abbreviated to P/T system) is a sextuple (S, T, F, G, K, W) where

- the quadruple (S, T, F, G) is a net and
- $K : S \to \aleph^{>0} \cup \infty$ and $W : S \times T + T \times S \to \aleph$ are functions such that for all $s \in S$ and $t \in T$ we have i) $W(s,t) > 0 \Leftrightarrow (s,t) \in F$ and ii) $W(t,s) > 0 \Leftrightarrow (t,s) \in G$.

The function K is called the *capacity* function of Ω and W is called the *weight* function of Ω. Note that suitably restricting W will yield two functions $W_F : F \to \aleph^{>0}$ and $W_G : G \to \aleph^{>0}$.

An *elementary net* is a P/T system (S, T, F, G, K, W) such that K is the constant function whose value is 1 and W is bounded above by 1 (in other words, the value of W is always less than or equal to 1).

The following definitions introduce the concepts needed to define behaviours of a P/T system $\Omega = (S, T, F, G, K, W)$.

A *marking* M of Ω is a function $M : S \to \aleph$ such that for all $s \in S$ we have the inequality $M(s) \leq K(s)$.

A subset $R \subseteq T$ of transitions is said to be *enabled* at a marking M if for all $s \in S$

$$\sum_{t \in R} W(s,t) \leq M(s) \leq K(s) - \sum_{t \in R} W(t,s). \tag{1}$$

(Of course, the empty sum equals $0 \in \aleph$.)

A marking M is said to have been *produced* from a marking L by the *occurrence* of $R \subseteq T$ if

- the marking L enables R and
- for all $s \in S$ we have the equality

$$M(s) = L(s) + \sum_{t \in R} (W(t,s) - W(s,t)). \tag{2}$$

Notice that requiring L to enable R guarantees that M is a marking. In the case of such an occurrence of R, it is usually said that the transitions $r \in R$ have *concurrently fired*.

The *state-space* of Ω is the reflexive graph $\mathcal{S}_\Omega$ defined as follows: its set of vertices is the set of markings of Ω; and an edge from L to M is a triple (L, R, M) where M is produced from L by the occurrence of $R \subseteq T$. The identity edge at M is $(M, \emptyset, M)$.

A behaviour of Ω is a finite path in $\mathcal{S}_\Omega$ – that is, a behaviour of the span $\mathcal{S}_\Omega : I \to I$ (as defined in the previous section).

Most often, P/T systems are given with a specified marking; this marking is interpreted as the initial state of the system. Of course, behaviours of such systems are paths beginning at the specified marking.

For any P/T system Ω we will construct a concurrent system Ψ_Ω such that the span $\mathcal{S}_\Omega : I \to I$ equals the value of Ψ_Ω.

First, we define a class of spans which will play the role of the places of a P/T system.

Let $\mathcal{N}$ be the one vertex reflexive graph whose set of edges is $\aleph$, the element $0 \in \aleph$ being the identity edge of $\mathcal{N}$.

Let U and V be finite sets and suppose $n \in \aleph^{>0} \cup \infty$. Define the reflexive graph $X_{U,V,n}$ as follows. Its set of vertices is $[n]$. An edge from the vertex $i \in [n]$ to the vertex $j \in [n]$ is a quadruple $(i, \mathbf{x}, \mathbf{y}, j)$ where $\mathbf{x} = (x_u)_{u \in U} \in \aleph^U$ and $\mathbf{y} = (y_v)_{v \in V} \in \aleph^V$ are tuples of natural numbers such that

$$\sum_{v \in V} y_v \leq i \leq n - \sum_{u \in U} x_u \tag{3}$$

and

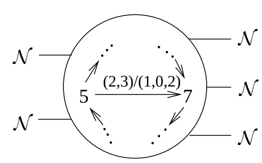
$$j = i + \sum_{u \in U} x_u - \sum_{v \in V} y_v. \tag{4}$$

We define the span $\mu_{U,V,n} : \mathcal{N}^U \to \mathcal{N}^V$ to be

$$\begin{tikzcd} & X_{U,V,n} \arrow[dl, "p"'] \arrow[dr, "q"] & \\ \mathcal{N}^U & & \mathcal{N}^V \end{tikzcd}$$

where $p(i, \mathbf{x}, \mathbf{y}, j) = \mathbf{x}$ and $q(i, \mathbf{x}, \mathbf{y}, j) = \mathbf{y}$.

We will draw these spans as circles. For example, the following is a depiction of an edge in such a system where $|U| = 2$ and $|V| = 3$.



Notice that if $|U| = 1$ and $|V| = 1$ then $X_{U,V,1}$ is (the head of) the message passer M defined in the previous section.

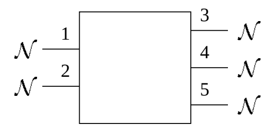
We now define a class of spans which will play the role of the transitions in a P/T system.

Let E be the one vertex graph with only one non-identity edge e. Given finite sets U and V as well as tuples of natural numbers $\mathbf{x} \in \aleph^U$ and $\mathbf{y} \in \aleph^V$, let $\tau_{U,V,\mathbf{x},\mathbf{y}} : \mathcal{N}^U \to \mathcal{N}^V$ be the span

$$\begin{tikzcd} & E \arrow[dl, "p"'] \arrow[dr, "q"] & \\ \mathcal{N}^U & & \mathcal{N}^V \end{tikzcd}$$

where $p(e) = \mathbf{x}$ and $q(e) = \mathbf{y}$.

When drawing such spans we will use squares and their boundaries will be labelled with the vectors $\mathbf{x}$ and $\mathbf{y}$. For example, if $\mathbf{x} = (1, 2)$ and $\mathbf{y} = (3, 4, 5)$ then we would draw $\tau_{U,V,\mathbf{x},\mathbf{y}}$ as follows.



Let $\Omega = (S, T, F, G, K, W)$ be a P/T system.

In order to facilitate describing Ψ_Ω we make the following definitions: for all $s \in S$ let $F(s, -) = \{t \in T \mid (s, t) \in F\}$ and $G(-, s) = \{t \in T \mid (t, s) \in G\}$; and for all $t \in T$ let $F(-, t) = \{s \in S \mid (s, t) \in F\}$ and $G(t, -) = \{s \in S \mid (t, s) \in G\}$.

For each place $s \in S$ let μ_s be the transition system

$$\mu_{G(-,s),F(s,-),K(s)} : \mathcal{N}^{G(-,s)} \to \mathcal{N}^{F(s,-)}.$$

For each transition $t \in T$ let $\mathbf{x} \in \aleph^{F(-,t)}$ be the tuple given by the function $F(-, t) \to \aleph : s \mapsto W_F(s, t)$ and let $\mathbf{y} \in \aleph^{G(t,-)}$ be the tuple determined by the function $G(t, -) \to \aleph : s \mapsto W_G(t, s)$. Define τ_t to be the transition system

$$\tau_{F(-,t),G(t,-),\mathbf{x},\mathbf{y}} : \mathcal{N}^{F(-,t)} \to \mathcal{N}^{G(t,-)}.$$

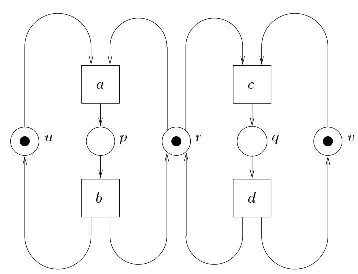
Let σ denote the permutation $\prod_{s\in S} \mathcal{N}^{F(s,-)} \cong \prod_{t\in T} \mathcal{N}^{F(-,t)}$ and let σ' denote the permutation $\prod_{t\in T} \mathcal{N}^{G(t,-)} \cong \prod_{s\in S} \mathcal{N}^{G(-,s)}$.

Let $n = |\sum_{s\in S} G(-, s)|$, $\eta = \eta_{\mathcal{N}^n} : I \to \mathcal{N}^n \otimes \mathcal{N}^n$ and $\epsilon = \epsilon_{\mathcal{N}^n} : \mathcal{N}^n \otimes \mathcal{N}^n \to I$. The concurrent system Ψ_Ω is defined to be the expression

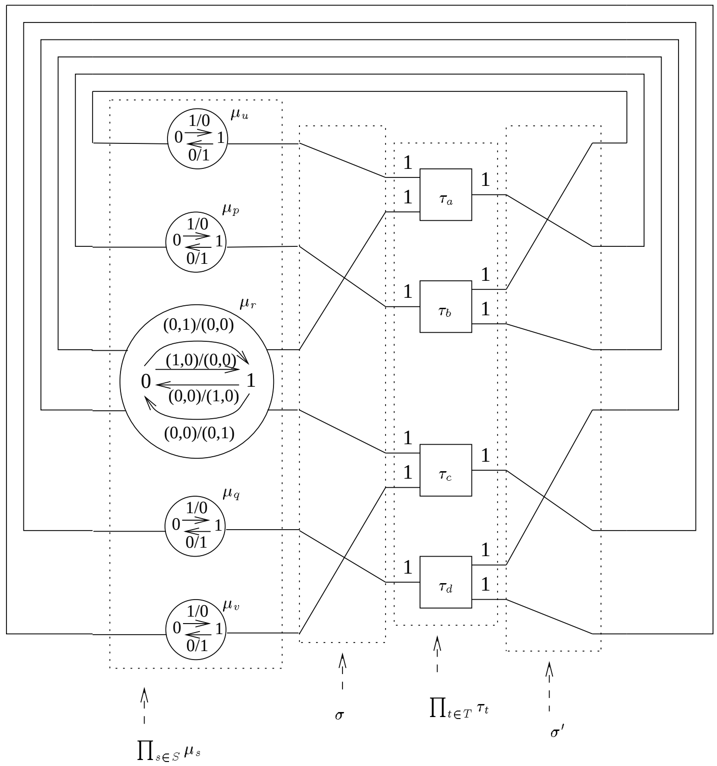
$$\eta; (\mathcal{N}^n \otimes ((\prod_{s\in S} \mu_s); \sigma; (\prod_{t\in T} \tau_t); \sigma')); \epsilon \ : \ I \to I.$$

Theorem *The state-space $\mathcal{S}_\Omega : I \to I$ is the value of the expression Ψ_Ω. Hence the behaviours of the net Ω and the concurrent system Ψ_Ω coincide.*

Before proving this theorem, let us consider an example. Below we depict the elementary net Ω (called mutex) which is often used to model a situation of mutual exclusion.



In this picture, the squares correspond to transitions and the circles to places. All the arcs are weighted by 1 and the dots represent the initial marking. The places u and v represent two users trying to gain access to the same resource r; when the place p is marked, u is using the resource, and when q is marked, v has access to it. We now draw the expression Ψ_Ω. (Some liberty has been taken in depicting the constants.)



The geometries of the two pictures are equivalent. That the picture for Ψ_Ω appears more complicated is because, in order to state a theorem such as the one above, the geometry of the system must be in a normal form. Of course, the picture of a P/T system has extra information in that it describes a marking – that is, a state of the system. For instance, the initial marking of mutex corresponds to the state $(u,p,r,q,v) = (1,0,1,0,1)$ of Ψ_Ω.

When reading the following proof, the reader may wish to particularize the argument to the above example.

Proof of Theorem To give a vertex of Ψ_Ω is to give a vertex of each component of the system. The only components which have more than one vertex are those of the form μ_s. Recall that the vertex-set of μ_s is $[K(s)]$. So to give a vertex of Ψ_Ω is to give a function $M : S \to \aleph$ such that $M(s) \leq K(s)$. This is, of course, a marking of Ω.

Recall that to give an edge of a concurrent system is to give a family of edges (e_c) – one edge for each component c of the system – which are compatible on the interfaces. Consider a candidate for an edge of Ψ_Ω; that is, a family of edges $e = (e_c)_{c\in S+T}$. Let $T(e) \subseteq T$ comprise those transitions $t \in T$ such that e_t is a non-identity edge; recall that (the head of) τ_t has one identity edge and one non-identity edge.

The claim is that a family of edges $e = (e_c)_{c\in S+T}$ defines an edge in Ψ_Ω from L to M if and only if M is produced from L by the occurrence of $T(e) \subseteq T$.

Suppose $e = (e_c)_{c\in S+T}$ defines an edge in Ψ_Ω from L to M. For each $t \in T(e)$ the edge e_t of $\tau_t : \mathcal{N}^{F(-,t)} \to \mathcal{N}^{G(t,-)}$ is labelled $((W(s,t))_{s\in F(-,t)}/(W(t,s))_{s\in G(t,-)})$. For all other $t \in T \setminus T(e)$, the edge e_t is the identity and hence labelled by only zeros.

Let $s \in S$. Recall that the edge $e_s \in \mu_s$ is labelled $\mathbf{x}/\mathbf{y} : L(s) \to M(s)$ where $\mathbf{x} = (x_t) \in \mathcal{N}^{G(-,s)}$ and $\mathbf{y} = (y_t) \in \mathcal{N}^{F(s,-)}$. The family of edges $e = (e_c)_{c\in S+T}$ must be compatible on the interfaces; therefore, $x_t = W(t,s)$ and $y_t = W(s,t)$ if $t \in T(e)$, while $x_t = y_t = 0$ if $t \in T\setminus T(e)$. So, $\sum_{t\in G(-,s)} x_t = \sum_{t\in T(e)} W(t,s)$ and $\sum_{t\in F(s,-)} y_t = \sum_{t\in T(e)} W(s,t)$. It is clear then that conditions 3 and 4 (given in the definition of μ) imply the conditions 1 and 2 ensuring that the occurrence of $T(e)$ produces M from L.

Now suppose that M is produced from L by the occurrence of $R \subseteq T$. Define a family of edges $e = (e_c)_{c\in S+T}$ as follows. For each $t \in T$ the edge $e_t \in \tau_t$ is the non-identity edge if and only if $t \in R$. For each $s \in S$, the edge e_s is labelled $(x_t)_{t\in G(-,s)}/(y_t)_{t\in F(s,-)}$ where $x_t = W(t,s)$ if $t \in R$, $x_t = 0$ if $t \in T\setminus R$, $y_t = W(s,t)$ if $t \in R$ and $y_t = 0$ if $t \in T\setminus R$. Conditions 1 and 2 guarantee that there is such an edge $e_s : L(s) \to M(s)$ in μ_s. It is straightforward to see this family of edges is compatible on the interfaces and that $T(e) = R$.

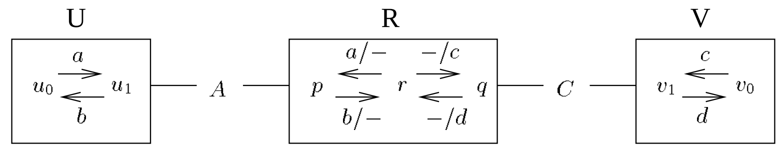
$\square$

Of course, there are expressions in **Span** which will not admit interpretations as P/T systems; for example, consider a single place $\mu_{U,V,n}$. It is easy, however, to identify a class of expressions for which such an interpretation is possible. Let Φ be a concurrent system comprising only spans $\mu_{U,V,n}$ (places) and $\tau_{U,V,\mathbf{x},\mathbf{y}}$ (transitions) such that each boundary of a place is in contact with the boundary of a transition and each boundary of a transition is in contact with the boundary of a place. It is clear that there is a net Ω such that the value of Ψ_Ω equals (actually, is isomorphic to) the value of Φ.

4 Final Remarks

Although we have shown how P/T systems can be constructed in a modular (that is, compositional) fashion from spans, we do not propose that **Span** be used to model concurrent systems in this way. The theory of Petri nets and the later developed P/T systems do not make a distinction between the places of a system and the states of a system – indeed, they are consciously blurred. In contrast to this, the calculus of spans does make a distinction between the *underlying geometry* of a system – which depicts the components of a system and their interfaces – and the *state-space* of each part of the system.

As an example, we would model the instance of mutual exclusion by the following expression Υ in **Span**. (The object A has non-identity edges a and b, and the object C has non-identity edges c and d.)



In the system there are three components: $U : I \to A$, $R : A \to C$ and $V : C \to I$. The interpretation of the states are as follows. When users U or V are respectively in states u_0 or v_0 they are not using the resource, while when in states u_1 or v_1 they do have access to it. When resource R is in state r, it is being used by no-one; when it is in state p, it is being used by some user connected to its left boundary – in Υ this user is U; and when R is in state q, it is being used by a user attatched to its right boundary – in this case, the user V. Here, a component plays the role of a place and an edge in the state-space of a component (which may have to synchronise on a boundary) plays the role of a transition. Notice that if Υ begins in state (u_0, r, v_0) then it has the same set of behaviours as mutex.

Compare the above picture with the net depicting mutex. The diagram of Υ has two levels of structure: one depicting the underlying geometry of the system (which is particularly simple) and the other, the state-spaces of the components. The picture of the net mutex can be thought of as a flattened version of the 'two-level geometry' depicting the concurrent system Υ. Also notice that the size of the state-space of (the value of) Υ is $2 \times 3 \times 2$, while the size of the state-space of mutex is 2^5.

The bicategorical structure of **Span** has not been discussed in this paper; it is, however, a fundamental part of the algebra's structure. In particular, it permits the development of a compositional theory of refinement and abstraction. This aspect of the theory will be treated by the authors elsewhere, wherein a comparison with bisimulation ([A93], [CS94], [JNW93]) will be given.

An important concept in net theory is that of hierarchy: roughly speaking, a hierarchical structure associates a net to a transition of a net. Though this is related to the notion of refinement mentioned in the preceeding paragraph, we would like express hierarchy in terms of the operations of our algebra. Given an expression in **Span**, built from the operations described in Section 2, we can replace any component of it by another expression. By doing this we are refining the underlying space of the system. We would like to do more: namely, refine both the underlying space as well as the state-space. This can be done by introducing more operations, the relevant operations coming from the algebra of *cospans* of graphs. (A cospan in a category $\mathbf{E}$ is a span in $\mathbf{E}^{\mathrm{op}}$.) An algebra which combines spans and cospans in order to express hierarchy will be defined in a future paper.

A span of graphs is a very abstract specification of a system, just as is a net. In modelling any real system one needs to specify quantities such as data-types. Two ways of achieving this in net theory are to use Predicate/Transition nets ([G86]) or coloured nets ([J86]). The bicategory **Span**, however, is closely related to the notion of a bicategory of processes in a distributive category, as defined in [KSW97a] – these processes in a distributive category being closely related to distributive automata ([SW93]) which have been constructed from data-types in a distributive category ([W92a],[W92b]). The way in which **Span** can be used to specify concurrent systems constructed from such data-types will also be described elsewhere.

5 Bibliography

[A93] Abramsky S., Interaction Categories (extended abstract), in: Theory and Formal Methods Workshop, Springer-Verlag (1993).

[A94] A. Arnold, Finite transition systems, Prentice Hall, 1994.

[B67] J. Bènabou, Introduction to bicategories, in: Reports of the Midwest Category Seminar, Lecture Notes in Mathematics 47, pages 1–77, Springer-Verlag, 1967.

[BSW96] Bloom S., Sabadini N, Walters RFC, Matrices, machines and behaviors, Applied Categorical Structures 4 (1996) 343-360.

[CS94] Cockett J.R.B. and Spooner D.A., SProc Categorically, in: Proceedings of CONCUR '94, Springer-Verlag (1994).

[CW87] A. Carboni and R.F.C. Walters, Cartesian Bicategories I, Journal of Pure and Applied Algebra, 49, pages 11-32, 1987.

[G86] Genrich H.J., Predicate/Transition nets, in: Petri nets, LNCS 254 Springer-Verlag (1986) 207-247.

[J86] Jensen K., Coloured Petri nets, in: Petri nets, LNCS 254 Springer-Verlag (1986) 248-299.

[JNW93] Joyal A., Nielsen M. and Winskel G., Bisimulation and open maps, in: Proceedings of the Eight Symposium on Logic in Computer Science, IEEE (1993).

[JS91] Joyal A. and Street R., The geometry of tensor calculus I, in: Advances in Math. 88 (1991) 55-113.

[JSV96] Joyal A., Street R. and Verity D., Traced monoidal categories, in: Math. Proc. Camb. Phil. Soc. 119 (1996) 447-468.

[K96] Katis P., Categories and bicategories of processes, PhD Thesis, University of Sydney (1996).

[KSW94] Katis P, Sabadini N, Walters RFC, The bicategory of circuits, Computing: Australian Theory Seminar, UTS, Sydney (1994).

[KSW97a] Katis P, Sabadini N, Walters RFC, Bicategories of processes, Journal of Pure and Applied Algebra, 115, no.2, pp 141 - 178, 1997.

[KSW97b] Katis P, Sabadini N, Walters RFC, Span(Graph): A categorical algebra of transition systems, Preprint, (1997).

[KL80] Kelly G.M. and Laplaza M.L., Coherence for compact closed categories, in: Journal of Pure and Applied Algebra 19 (1980) 193-213.

[M70] S. Mac Lane, Categories for the working mathematician, Springer Verlag, 1970.

[M89] Milner R., Communication and Concurrency, Prentice Hall International (1989).

[MM90] Meseguer J. and Montanari U., Petri nets are monoids, in: Information and Computation, 88 (2) 105-155 (1990).
[MSSW95] Mauri G., Sabadini N., Shammah S. and Walters R.F.C., On distributive automata, asynchronous automata and Petri nets, Preprint, 1995.
[P71] R. Penrose, Applications of negative dimensional torsors, in Combinatorial Mathematics and its applications, (D. J. A. Welsh, Ed.) pp. 221-244, Academic Press, New York, 1971.
[R86] Reisig W., Place/Transition systems, in: Petri nets, LNCS 254 Springer-Verlag (1986) 117-141.
[SW93] Sabadini N. and Walters R.F.C., On functions and processors: an automata-theoretic approach to concurrency through distributive categories, School of Mathematics and Statistics Research Reports, University of Sydney, (93-7), 1993.
[SWW93] Sabadini N., Walters R.F.C., Weld Henry, Distributive automata and asynchronous circuits, Category Theory and Computer Science 5, Amsterdam, 28-32, 1993.
[SW93] N. Sabadini and R.F.C. Walters, On functions and processors: an automata-theoretic approach to concurrency through distributive categories, School of Mathematics and Statistics Research Reports, University of Sydney, (93-7), 1993.
[T86] Thiagarajan P. S., Elementary net systems, in: Petri nets, LNCS 254 Springer-Verlag (1986) 26-59.
[W92a] R.F.C. Walters, Categories and Computer Science, Carslaw Publications 1991, Cambridge University Press 1992.
[W92b] R.F.C. Walters, An imperative language based on distributive categories, Mathematical Structures in Computer Science, 2:249–256, 1992.