Shallow Linear Action Graphs and their Embeddings

James J. Leifer and Robin Milner
University of Cambridge, Computer Laboratory
October 2000

In previous work, action calculus has been presented in terms of *action graphs*. Many calculi, or at least their salient features, can be expressed as specific action calculi; examples are Petri nets, λ-calculus, π-calculus, fusion calculus, ambient calculus and spi calculus.

We here offer *linear* action graphs as a primitive basis for action calculi. Linear action graphs have a simpler theory than the non-linear variety. This paper presents the category of embeddings of *shallow* linear action graphs (those without nesting), using a novel form of graphical reasoning which simplifies some otherwise complex manipulations in regular algebra. The work is done for undirected graphs, and adapted in a few lines to directed graphs.

The graphical reasoning used here will be applied in future work to develop behavioural congruences for action calculi.

1 Introduction

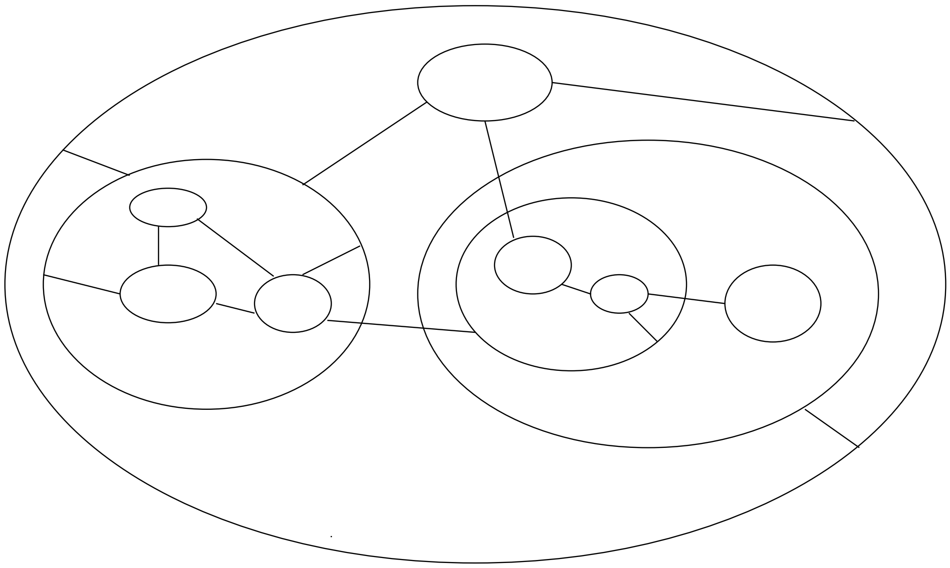
This paper is part of long-term project, which began in about 1991, to develop a general mathematical model of mobile computing which can underpin many different formulations. The action graphs [14], with which the project began, admit great freedom in the way in which graphs can reconfigure themselves, reflecting different aspects of mobility. That first phase of work has also revealed the need to understand a more primitive form of graph: those in which the wiring is *linear* in a sense to be described. Their theory will be simpler, and the exploratory work we have done indicates that non-linear varieties can be recovered in terms of the linear ones. The present paper is therefore the first of a sequence of several which will lay out the linear and non-linear theories in careful progression.

Graphical reasoning is prominent in this paper. For understanding certain aspects of action graphs we find it essential to use a kind of *graphical formulae*, i.e. assertions expressed in graphical notation. Other authors have found graphical forms of assertion useful in computation theory. A notable example is Burstall's 1972 paper [2], a pioneering use of graphical reasoning to enlighten the hazardous business of updating list structures by swinging pointers. His work is indeed a rigorous treatment of mobility in *data structures*; our present work can be seen as a generalisation of Burstall's ideas, in order to deal with mobility of *processes*, for example in the "global computer" which the internet is rapidly becoming.

Background Action calculus [14] is an approach to a general model for mobile interactive computing. An individual action calculus $\mathsf{AC}(\mathcal{K})$ is determined by its set $\mathcal{K}$ of *controls* and their associated *reaction rules*. Petri nets, the λ-calculus and the π-calculus are all action calculi. There is also a growing range of more modern calculi, mostly focussed upon specific features of mobile distributed computing; examples are the spi calculus [1] and the ambient calculus [3]. These, or at least many of their features, fall within the action calculus framework. By a familiar abuse of language, we use the term "action calculus" to mean either this framework or any individual calculus that falls within it.

What is the use of a framework for calculi? If calculi share common material, in particular if they can be formulated in action calculus, then they may also benefit from shared theory. There are other shared theories in computing; for example, the analysis of programming languages within a particular semantic or type-theoretic framework. Semantics of programming has alleviated the threat of a Babel of languages; the threat of a Babel of calculi is now upon us, and we would like to alleviate that threat too. If action calculus is to help, then it must provide shared theory, not just shared formalism.

The first steps [14] towards a shared theory of action calculi rested on their presentation using *action graphs*. An essential feature of these graphs, not conventional in graph theory, is their *nesting*; each node may contain a *child* graph, and nesting may be arbitrarily deep. Nesting represents both *scope*, e.g. of a name in the π-calculus, and *location*, e.g. a site of activity such as an ambient of the ambient calculus. Here is a picture –deliberately vague in detail– of such a graph:



Here we have allowed a *node*, at any depth, to be linked by an *edge* to any other node. The crucial element of each action calculus is its reaction rules, each re-configuring a subgraph; the rules will model message-passing, migration, replication, destruction or any other mode of interaction. Thus, communication in the π-calculus is modelled by edge manipulation, and guarding –e.g. the prefix $x(y).P$– by encapsulation of P within a node, whose boundary is destroyed when the prefix reacts. (In general, when a node boundary is destroyed, the rule must determine where to link edges which impinge on the boundary.) Movement in the ambient calculus is modelled quite directly; an ambient is one kind of node, edges represent references to ambients, and the reaction rules specify the movement of ambients across ambient boundaries and the creation or removal of such boundaries. These two examples illustrate different reaction disciplines; the first inhibits activity inside certain nodes, while the second permits it.

In [14] a particular class of action graphs is treated algebraically. Some of its features are as follows: Graphs are the arrows of a strict symmetric monoidal category whose objects are arities; edges are directed, and impinge on nodes at *source* and *target* ports; moreover edges can fork –many can emerge from a single source port– but not join at a target port; edges may only cross node boundaries going inwards (from source to target); and so on. These details are mentioned only to illustrate the possible variety; other choices have been taken, for example by Honda [8] and Gardner and Wischik [6]. At the end of this section is a taxonomy which is incomplete, but enough to place our present work in relation to the varieties mentioned.

Purpose A long-term aim is to provide a rich behavioural theory of such graphical models, since we believe that a combination of rigour and geometric intuition is essential to the engineering of mobile interactive systems. To begin with, we clearly have to understand "behaviour"; what does the term mean? Guided by experience with process calculi, we approach this question via a study of behavioural preorders or equivalences based directly upon reaction rules.

Recently we have begun to find a behavioural theory, common to many action calculi. We defined a general notion of *reactive system*, which includes action graphs and their reaction rules. We then showed [12, 11] that, for certain "nice" reactive systems, a labelled transition system (LTS) can be derived whose associated preorders and equivalences –such as trace inclusion, trace equivalence, the failures ordering and (bi)simulation– are congruential. Then in [5], as an exploratory exercise, we showed that without nesting a wide range of categories of action graphs and their embeddings are "nice" reactive systems. We believe that this result can be extended to admit nesting; at that point, we shall be in a position to compare our general behavioural theory with what has been previously developed independently for many real calculi, including the π-calculus.

This work makes essential use of graphical properties. But graphical formulation levies a toll! The conditions on reaction rules and graph embeddings which ensure "nice" reactive systems are not at all obvious. We first have to set up an appropriate category of action graphs under embeddings, and this turns out to be subtle. Moreover, the embeddings in our exploratory study [5] admit only a constrained form of *reflexion* or feedback (see Section 3), and for a satisfactory theory we would like no constraint on reflexion. One of us (Leifer) has indeed adapted the axioms to this end; the justifying proofs, requiring considerable use of regular algebra, will appear in his forthcoming PhD dissertation [11].

Taking into account the difficulties already encountered, before extending our theory we seek to base it on a more primitive foundation. We therefore study *linear* action graphs here, expecting so to shed more light upon the non-linear case. Indeed, linear graphs are close to previously-studied structures, at least in the shallow (non-nesting) case; Cattani [4] points out, relying on Mifsud's work [13], that they constitute the free *traced strict symmetric monoidal category* over a signature $\mathcal{K}$ of controls. Joyal et al [9], deploying techniques of graphical reasoning, have axiomatised these categories. But this does not determine how to handle embeddings of linear graphs.

The present paper sets up the category of embeddings for shallow linear action graphs, and verifies the construction using a novel form of diagrammatic reasoning that does the job for which Leifer [11] used regular algebra. The technique has wider applicability; in a sequel to this paper we shall use it to establish the conditions under which we get the "niceness" mentioned above.

Taxonomy of graphs There are many varieties of action graph. They can be linear or non-linear; directed (see Section 2) or undirected; shallow or deep; closed (no freely-named ports) or open. There are also varieties of embedding – e.g. (as mentioned) they can admit more or less of reflexion, and (for deep graphs) can be deep or shallow. A full taxonomy would be complex, since these degrees of freedom are not always orthogonal. We shall merely distinguish a few, guided by what we consider important and by how we predict the theory will develop.

We believe that the category **G-Emb** of *linear, undirected, deep, open* action graphs and appropriate embeddings will work well in applications, and that its theory will underpin the theories of other varieties. For example, we expect to recover non-linear varieties [14, 6, 8] by a suitable quotient of the linear category **G-Emb**. We expect also to derive the associated *directed* variety, and to consider *shallow* and *closed* varieties.

So we shall subscript **G-Emb** with any of 's', 'c', 'd' to mean combinations of shallow, closed and directed; e.g. $\textbf{G-Emb}_{\mathsf{sc}}$ for the shallow, closed but undirected variety. We also expect to introduce subscripts for various non-linear forms.[1]

In each case we expect (at first) to study fully reflexive embeddings of a single graph in another. Later, following Sewell's [16] study of multi-hole contexts, we shall want to consider multiple embeddings, and may also vary embeddings in other ways; we may wish to use superscripts to classify varieties of embedding.[2]

The present paper establishes the category $\textbf{G-Emb}_{\mathsf{s}}$ of shallow undirected linear action graphs, and derives $\textbf{G-Emb}_{\mathsf{sd}}$, its directed variant. The prime instance of $\textbf{G-Emb}_{\mathsf{s}}$ is Lafont's interaction nets [10]; these are also generalised by Turnbull [17], whose forthcoming PhD thesis [17] will propose a linear intermediate language for interactive programming.

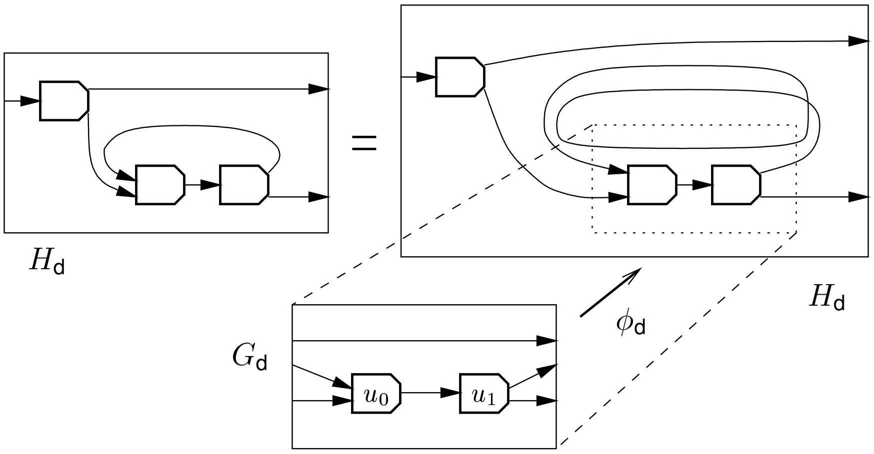
2 Linear action graphs

In this section we shall define the category $\textbf{G-Emb}_{\mathsf{s}}$ of shallow undirected linear action graphs and embeddings. This depends on the subsidiary category **E-Emb** of *edge embeddings* which is detailed in Section 3.

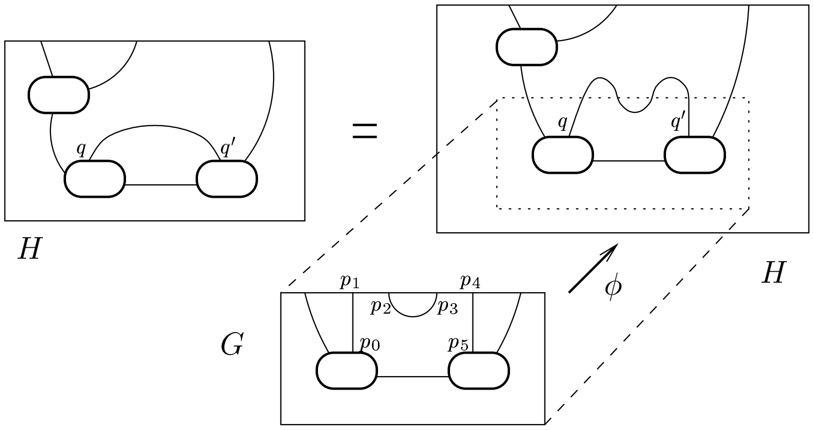
Before making precise definitions, let us explain with diagrams the kind of embedding that we want to understand. Here are two *directed* action graphs G_{d} and H_{d}, of the kind studied in [14, 5], together with an embedding ϕ_{d} of G_{d} into H_{d}. Each node u can carry *ports*; *targets* at its left and *sources* at its right. Also a graph may have both *input* sources at its left and *output* targets at its right.

[1]The name $\textbf{G-Emb}_0$ was used in [5] for a category of shallow, closed, directed, *non-linear* action graphs. That name does not accord with the convention proposed here.

[2]Surely, only a few of the possible combinations of graph and embedding will be interesting! We have no intention of drowning in a sea of circumscripts – we merely want to avoid confusion in navigating the archipelago of varieties.



The diagram shows how G_{d} can be "cut out" of H_{d}. It also shows the graphical *context* which remains if we remove from H_{d} the copy of G_{d} inside the dotted line. There is a close connection between such contexts and embeddings; in fact there are several different contexts into which G_{d} may be inserted to yield H_{d}, and we want to reflect this difference in our definition of embedding. The diagram illustrates a use of *reflexion* in the embedding; an outgoing edge from node u_1 is bent back to form an ingoing edge of u_0, and this bending also cycles once more through the input-output edge of G_{d}. The example illustrates how a graph and a context can collaborate to create a link.



The second diagram shows an undirected version of this situation (ignore the port labels p_i for now). G and H are derived from G_{d} and H_{d} simply by deleting directional information; thus, source ports and target ports are no longer distinguished, and we line them up along the top edge of the graph. But we want to reverse this derivation, for we find that many aspects of our theory are best developed for undirected graphs first, and then transferred to directed graphs. In Section 4 we shall make precise how directed graphs and their embeddings arise from *imposing direction* on undirected ones. Direction can be imposed in many ways on a graph; in the present example it can be imposed on G and H to obtain G_{d} and H_{d} in a way which is respected by the embedding ϕ, and thus the embedding ϕ_{d} is recovered.

Terminology If R is a binary relation then $(y, x) \in R$, yRx, $y \in Rx$ and $xR^{-1}y$ all mean the same; so does $y = Rx$ if R is a partial function. (In this work there is no risk of confusing a singleton set with its member.) The composition of two relations is defined by $z(R' \circ R)x \overset{\text{def}}{\Leftrightarrow} \exists y.\ zR'y \,\&\, yRx$. The *domain* of R is $\mathsf{dom}\, R = \{x \mid \exists y.\ yRx\}$ and its *range* $\mathsf{ran}\, R = \{y \mid \exists x.\ yRx\}$. We denote the transitive closure of R by R^+, and its transitive reflexive closure by R^*. We understand the application $R' R\, y$ to mean $R'(R(y))$.

We write $A \uplus B$ for the union of two sets A and B known or assumed to be disjoint. We write Id_A for the identity function on a set A. We write $f : A \rightleftarrows B$ for a bijection, and $f : A \rightleftharpoons B$ for a partial bijection, or equivalently a partial injection – i.e. a partial function which, restricted to its domain, is injective. We shall call a bijection $f : A \rightleftarrows A$ an *involution* if it is self-inverse, i.e. $y = fx \Rightarrow x = fy$; a *strict* involution f is one for which $\forall x.\ fx \neq x$. A *partial* (strict) involution is a partial function which, restricted to its domain, is a (strict) involution.

In categories, we write the composition of two arrows $f : A \to B$ and $g : B \to C$ as $g\, f : A \to C$, and the identity arrow on A as id_A. We write $[n]$ for the ordinal set $\{0, 1, \ldots, n-1\}$.

We now proceed to define *edgings* and *edge embeddings*, on which our definitions for graphs depend.

Definition 1 (edging) A *(linear, undirected) edging* is a pair $E = (P, edg)$, where edg is strict involution on the set P of *ports*. If $p' = edg\, p$ we call $\{p, p'\}$ an *edge*, and write $\{p, p'\} \in edg$. ∎

Later we shall define directed edgings; their properties can often be derived from those of undirected ones. We shall use $E = (P, edg_E)$, E_0, E', $F = (Q, edg_F)$, $\ldots$ to denote edgings, and understand that their components are similarly subscripted or primed.

Definition 2 (graphs) A *(shallow, linear) action graph* takes the form $G = (L, U, ar, edg)$, where L is a finite set of *external* ports[3], U is a finite set of *nodes*, the function $ar : U \to \mathbb{N}$ provides the *arity* of each node, and $E = (P, edg)$ is a linear, undirected edging on the port set

$$P \overset{\text{def}}{=} \sum_{u \in U} [ar(u)] + L\ . \qquad ∎$$

We shall denote the injections induced by the disjoint sum P as follows:

$$\begin{array}{rll} bind(u) : & [ar(u)] \rightarrowtail P & \text{for the binding ports of each } u \in U; \\ ext : & L \rightarrowtail P & \text{for the external ports.} \end{array}$$

For any injection i into a set A we may write A^i for the range of i; thus P^{ext} is the set of all external ports and $P^{bind(u)}$ the set of binding ports of the node u. We shall also write for example P^{bind} for $\biguplus_{u \in U} P^{bind(u)}$. With this notation we see that $P = P^{bind} \uplus P^{ext}$.

[3]External ports include the input and output ports of the directed graphs studied in [5], and also free names; in this paper it suffices to treat them together. In [5] we also ascribed a *control* $K \in \mathcal{K}$ to each node; for brevity we omit controls here, but the results of this paper extend easily to admit them.

If an edge $\{p, p'\}$ is a subset of P^{ext} we shall call it a *link edge*. For the rest of this section, Section 3 and Section 4 we shall omit the terms "shallow" and "linear", since they apply to all our constructions.

Definition 3 (edge embedding) Let $E = (P, edg_E)$ and $F = (Q, edg_F)$ be edgings. An *edge embedding* $\phi : E \to F$ is a pair $(\phi^{\mathsf{P}}, \phi^{\mathsf{R}})$, where the *port map* $\phi^{\mathsf{P}} : P \rightleftharpoons Q$ is a partial bijection, and the *reflexion map* ϕ^{R} is a partial strict involution on P. We also write $\{p, p'\} \in \phi^{\mathsf{R}}$ for $p' = \phi^{\mathsf{R}} p$. Two conditions are imposed:

$$\begin{array}{ll} \text{E-DOM}: & \mathsf{dom}\,\phi^{\mathsf{P}} \uplus \mathsf{dom}\,\phi^{\mathsf{R}} = P \\ \text{E-EDGE}: & edg_F \circ \phi^{\mathsf{P}} \subseteq \phi^{\mathsf{P}} \circ (edg_E \circ \phi^{\mathsf{R}})^* \circ edg_E\,. \end{array}$$
■

Thus, E-DOM asserts that $\phi^{\mathsf{P}} \cup \phi^{\mathsf{R}}$ is a total function. The effect of E-EDGE, as we shall see, is to ensure that a chain of edges in E linked by reflexion either embeds into a single edge of F or forms a reflexive loop. It is not quite trivial to see this; in fact, proofs about this simple definition of embedding involve tricky regular algebra. To avoid this, in the next section we shall introduce diagrams called *embedding formulae*, which will make the proofs both shorter and more intuitive. For now, consider the embedding $\phi : G \to H$ illustrated above. Among the edges of G are $\{p_0, p_1\}$,$\{p_2, p_3\}$ and $\{p_4, p_5\}$, as shown. Since the ports p_0 and p_5 "appear" in H as q and q' we have

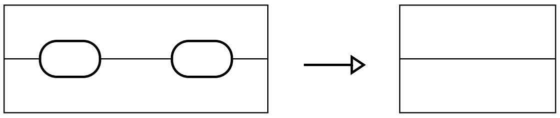
$$\phi^{\mathsf{P}} p_0 = q, \;\; \phi^{\mathsf{P}} p_5 = q'\,.$$

But the embedding of the three mentioned edges into the edge $\{q, q'\}$ is represented also by two members of ϕ^{R}, namely

$$\{p_1, p_2\} \in \phi^{\mathsf{R}}, \;\; \{p_3, p_4\} \in \phi^{\mathsf{R}}\,.$$

We think of this as the embedding into H of a chain $p_0, p_1, p_2, p_3, p_4, p_5$ whose consecutive pairs are alternately edges of G and reflexions of ϕ^{R}.

This chain formation is what makes our construction of **G-Emb**$_{\mathsf{S}}$ non-trivial, so we should pause to consider why we need it for our model. Certainly, conventional graphs do not contain link edges, such as $\{p_2, p_3\}$ in our illustration graph G. One reason we need them is to admit a reaction rule of the form



which replaces two linked nodes by an edge, and thus may create a link edge in any graph to which it is applied. Such rules arise naturally. Furthermore, link edges are also important in *contexts*. A simple example arises even in sequential programming; a program context which makes one program variable an alias for another may be modelled naturally by a graph context with a link edge.

We are now ready to define the embedding of one graph into another.

Definition 4 (graph embedding) Let $G = (L, U, ar_G, edg_G)$ and $H = (M, V, ar_H, edg_H)$ be two action graphs, with $E = (P, edg_G)$ and $F = (Q, edg_H)$. Then a *graph embedding* $\phi : G \to H$ is a pair $\phi = (\phi^{\mathsf{V}}, \phi^{\mathsf{E}})$ where $\phi^{\mathsf{V}} : U \rightarrowtail V$ is an injection of nodes, and $\phi^{\mathsf{E}} = (\phi^{\mathsf{P}}, \phi^{\mathsf{R}}) : E \to F$ is an edge embedding which respects arity and binding, i.e.

$$\begin{array}{ll} \text{E-AR}: & ar_H \circ \phi^{\mathsf{V}} = ar_G \\ \text{E-BIND}: & bind_H(\phi^{\mathsf{V}} u) = \phi^{\mathsf{P}} \circ (bind_G\, u) \quad . \end{array}$$ ■

This implies that ϕ^{P} is defined, and hence ϕ^{R} undefined, on all binding ports of G.

In Section 3 we shall define the composition of edge embeddings, and thereby establish in Theorem 15 a category of edge embeddings. Relying on that result, we now define the category of graph embeddings.

Definition 5 (category of graphs and embeddings) The category **G-Emb**$_{\mathsf{s}}$ has graphs as objects and graph embeddings as arrows. The composition of arrows is given by $\psi\phi \stackrel{\text{def}}{=} (\psi^{\mathsf{V}} \circ \phi^{\mathsf{V}}, \psi^{\mathsf{E}}\phi^{\mathsf{E}})$, and the identity embedding id_G on $G = (L, U, ar, edg)$ with edging $E = (P, edg)$ is given by $\mathsf{id}_G = (\mathsf{Id}_U, \mathsf{id}_E)$. ■

Theorem 6 **G-Emb**$_{\mathsf{s}}$ *is a category.*

Proof It is clear that the composition of two embeddings is indeed an embedding, from Prop. 10 and the easy fact that E-AR and E-BIND are satisfied by the composite. In proving composition associative and establishing the identities, the only non-trivial work is done by Prop. 13. ■

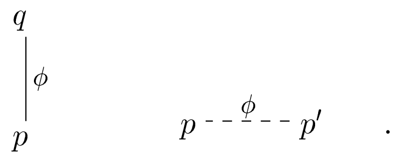
3 Linear edgings

In this section, starting from Def. 1 of edging and Def. 3 of edge embedding, we shall show that undirected edgings and embeddings form a category which we call **E-Emb**. This is not immediate. Its subtlety is hinted by the illustration at the beginning of Section 2, showing how edges can be created by co-operation between a graph and its context. To express the composition of embeddings, and to verify that this yields a category, we find it helpful to use a special form of diagrammatic reasoning.

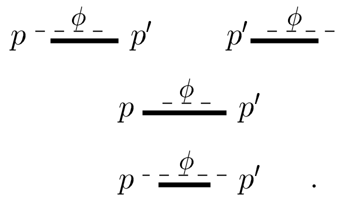
We therefore introduce some diagrams. Each diagram makes an assertion about the ports and embeddings mentioned in it; we call these diagrams *embedding formulae*. First, the assertion that $p = edg\, p'$, i.e. that $\{p, p'\}$ is an edge, is drawn as

$$p \;\rule[0.5ex]{3em}{1.5pt}\; p' \quad .$$

(Naming conventions will make it clear which edging it belongs to.) Since edg is an involution, to swap p and p' would not change the meaning. Next, given an edge embedding $\phi : E \to F$, we assert that $q = \phi^{\mathsf{P}}\, p$ and $\{p, p'\} \in \phi^{\mathsf{R}}$ respectively by

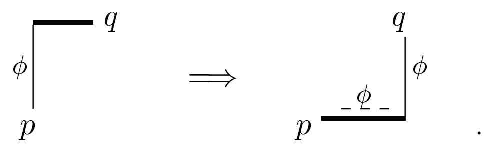


The latter diagram is called a *ϕ-reflexion*, and again has left-right symmetry. Our main elementary embedding formula, called a *ϕ-reflexion chain*, represents an alternating sequence of zero or more edges and ϕ-reflexions, where each two adjacent links in the chain share a port and the end ports are p and p'. There are four forms of the diagram:

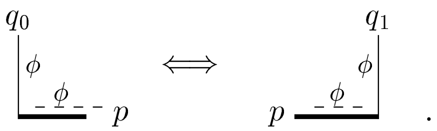


The overlaps of dashed and bold lines in each diagram describe the ends of the chain. The first two diagrams mean the same; they both assert $p\,(\phi^{\mathsf{R}} \circ edg)^*\, p'$, i.e. the chain may be empty or else the leftmost (resp. rightmost) link is a reflexion and the rightmost (resp. leftmost) is an edge. The third is an instance of $edg \circ (\phi^{\mathsf{R}} \circ edg)^*$, or equivalently $(edg \circ \phi^{\mathsf{R}})^* \circ edg$, i.e. the end links are both edges; the fourth is an instance of $(\phi^{\mathsf{R}} \circ edg)^* \circ \phi^{\mathsf{R}}$, i.e. the end links are both reflexions.

These elementary formulae can be combined to represent complex assertions. At each join, the omission of a port name implies that it is existentially quantified. For example, the antecedent and consequent of the condition E-EDGE are expressed by compound diagrammatic formulae:

E-EDGE : 

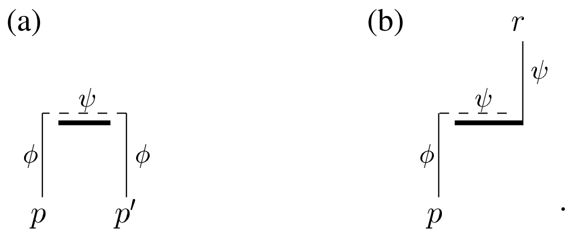
Proposition 7 *Let $\phi : E_0 \to E_1$ be an edge embedding, and $q_0 \;\boldsymbol{-\!\!-\!\!-}\; q_1$ an edge of E_1. Then for all $p \in P$*



Proof Assume the lefthand side, with p_0 such that $q_0 = \phi^{\mathsf{P}} p_0$. By E-EDGE there exists p_1 such that $q_1 = \phi^{\mathsf{P}} p_1$, and since ϕ^{P} is injective $p_0 \overset{\phi}{\boldsymbol{-\!\!-\!\!-}} p_1$. The reflexion chain starting from p_0 is uniquely determined, and by E-DOM cannot proceed beyond p_1, hence meets p en route; so $p \overset{\phi}{\boldsymbol{-\!\!-\!\!-}} p_1$, and the righthand side follows. The reverse implication is similar. ∎

For the rest of this section, in discussing embeddings between edgings E_0, E_1, E_2, E_3, we shall consistently use the letters p, q, r, s to denote members of their respective port sets. We first consider how to define the composition of two edge embeddings $\phi : E_0 \to E_1$ and $\psi : E_1 \to E_2$. We have to define a partial bijection $(\psi\phi)^{\mathsf{P}} : P_0 \rightleftharpoons P_2$ and a partial strict involution $(\psi\phi)^{\mathsf{R}} : P_0 \rightleftharpoons P_0$. We first describe them informally, in a way which clearly secures E-DOM for the composite $\psi\phi$.

Start with $p \in P_0$. If $p' = \phi^{\mathsf{R}} p$ is defined then we take $\{p, p'\} \in (\psi\phi)^{\mathsf{R}}$, leaving $(\psi\phi)^{\mathsf{P}} p$ undefined. Otherwise let $q = \phi^{\mathsf{P}} p$; starting from q with a reflexion, develop a ψ-reflexion chain until encountering a q' either (a) in the *range* of ϕ^{P}, say $\phi^{\mathsf{P}} p' = q'$, or (b) in the *domain* of ψ^{P}, say $\psi^{\mathsf{P}} q' = r$ (indeed, if $\psi^{\mathsf{P}} q$ itself is defined then the chain is empty and $q' = q$). Exactly one of (a) and (b) must occur first, for if the chain revisits q – which is in the range of ϕ^{P} – then by E-EDGE it has already encountered $q' = edg_1\, q$ in the range of ϕ^{P}.



In case (a) we take $\{p, p'\} \in (\psi\phi)^{\mathsf{R}}$, and in case (b) we take $(\psi\phi)^{\mathsf{P}} p \overset{\text{def}}{=} r$; in each case the other map is left undefined on p. Thus E-DOM is secured, and with the exception of E-EDGE it follows easily from our construction that $(\psi\phi)^{\mathsf{P}}$ and $(\psi\phi)^{\mathsf{R}}$ have the properties required for an edge embedding.

It remains to check that E-EDGE is satisfied by this construction, but first we shall introduce a refinement of our diagrams to state the construction succinctly. We refine the elementary edge assertion to the following

$$q \underset{\phi}{\rule{3em}{1pt}} q' \quad ,$$

which we call a *ϕ-free edge (formula)*; it asserts not only that $\{q, q'\}$ is an edge but also that $q, q' \notin \mathsf{ran}\, \phi^{\mathsf{P}}$. We also refine the notion of reflexion chain to a *ψ/ϕ-reflexion chain*

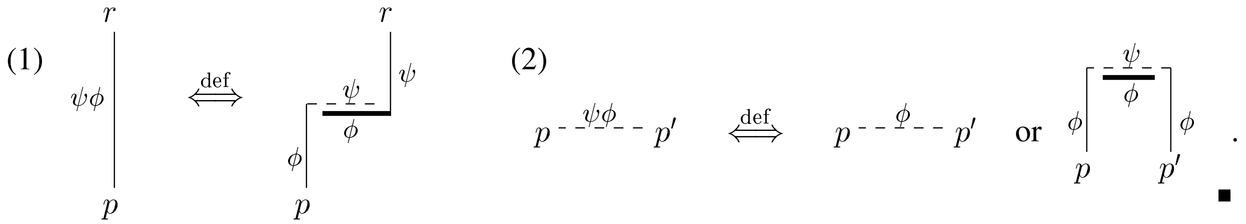
$$q \overset{\psi}{\underset{\phi}{\rule{3em}{1pt}}} q' \quad , \quad q \overset{\psi}{\underset{\phi}{\rule{3em}{1pt}}} q' \qquad \text{etc.}$$

each of which asserts that there is a ψ-reflexion chain from q to q', and moreover that every edge on the chain is ϕ-free. This only constrains *edges* in the chain; it does not constrain an end port of the chain if the end link is a reflexion.

We shall see shortly how this diagrammatic convention represents a partial duality, hinted by the fact that each ψ-reflexion $q' = \psi^{\mathsf{R}} q$ implies $q \notin \mathsf{dom}\, \psi^{\mathsf{P}}$ (or equivalently $q' \notin \mathsf{dom}\, \psi^{\mathsf{P}}$). This "up-down" duality should not be confused with the left-right symmetry of our diagrams.

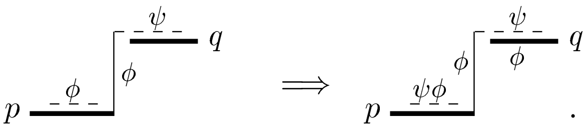
With our refined embedding formulae we can now define composition of embeddings entirely by diagrams:

Definition 8 (composition of embeddings) Let $\phi : E_0 \to E_1$ and $\psi : E_1 \to E_2$ be two edge embeddings. Then the composite embedding $\psi\phi : E_0 \to E_2$ is defined by the following diagrams:

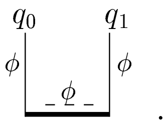


To complete our proof that this is a good definition we need a lemma:

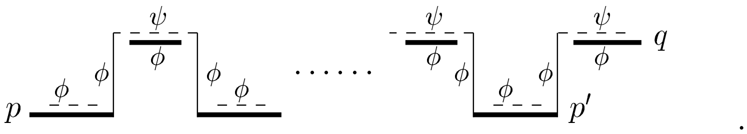
Lemma 9



Proof By E-EDGE, any edge $\{q_0, q_1\}$ in the antecedent's ψ-reflexion chain which is in the range of ϕ can be replaced by the $\sqcup$-configuration



This leaves only ϕ-free edges; so from the antecedent we deduce

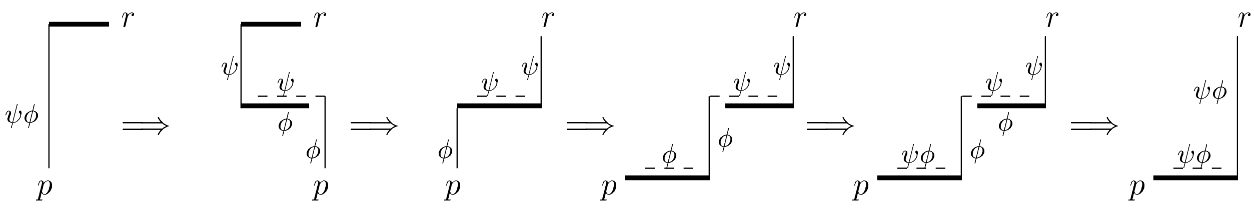


Now by Def. 8(2) each $\sqcap$-configuration in this formula implies a $\psi\phi$-reflexion, and so does each ϕ-reflexion; hence the part of the formula between p and p' implies a $\psi\phi$-reflexion chain, and the consequent follows. ∎

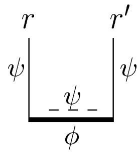
We are now in a position to justify Def. 8.

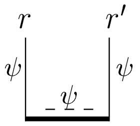
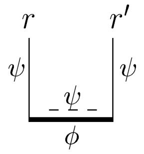
Proposition 10 *The composition of two edge embeddings is an edge embedding.*

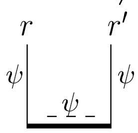
Proof We have already argued that $(\psi\phi)^{\mathsf{P}}$ and $(\psi\phi)^{\mathsf{R}}$ as defined in Def. 8 have the right properties, except for E-EDGE. The proof that E-EDGE is satisfied is now brief. In the first step below we use left-right symmetry in unpacking Def. 8(1)), and in the second step before applying Prop. 7 we weaken a ψ/ϕ-reflexion chain to a ψ-reflexion chain:



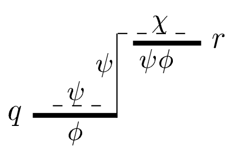
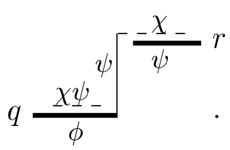
The main task in showing that edgings and embeddings form a category is to show composition associative. For this, we need a property which is interesting in its own right; as promised, it is a kind of dual of Def. 8(2).

Proposition 11 $r \frac{\qquad}{\psi\phi} r' \iff r \frac{\qquad}{\psi} r'$ or  .

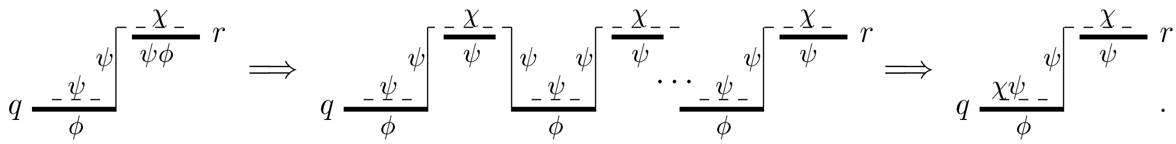
Proof ($\Longrightarrow$) Assume the lefthand side. If $\{r, r'\}$ is a ψ-free edge, then $r \frac{\qquad}{\psi} r'$. If not, then E-EDGE yields  ; furthermore every edge on this ψ-reflexion chain must be ϕ-free, else r would be in the range of $(\psi\phi)^{\mathsf{P}}$ contradicting the assumption, hence  .

($\Longleftarrow$) Assume the righthand side. If the first disjunct holds, i.e $r \notin \mathsf{ran}\,\psi^{\mathsf{P}}$, then by Def. 8(1) $r \notin \mathsf{ran}(\psi\phi)^{\mathsf{P}}$ and we are done. If the second holds, then the defining formulae for $(\psi\phi)^{\mathsf{P}}$ in Def. 8(1) cannot hold for r and any p; hence $r \notin \mathsf{ran}(\psi\phi)^{\mathsf{P}}$. Also $\{r, r'\}$ must be an edge; for, given r, the port r' is unique such that  holds, and by E-EDGE it certainly holds when $r' = edg_2 r$. Hence $r \frac{\qquad}{\psi\phi} r'$. ∎

Thinking (as we do) of embeddings going upwards, this duality seems to say that reflexion links look the same "from on top" as edge links look "from underneath". We now show that the duality yields an elegant zigzag property, as follows:

Lemma 12  $\iff$  .

Proof ($\Longrightarrow$) In the χ-reflexion chain, replace each $\psi\phi$-free edge by either a ψ-free edge or a $\sqcup$-configuration, as justified by Prop. 11. Then use Def. 8(2) to replace each $\sqcap$-configuration and each ψ-reflexion by a $\chi\psi$-reflexion.

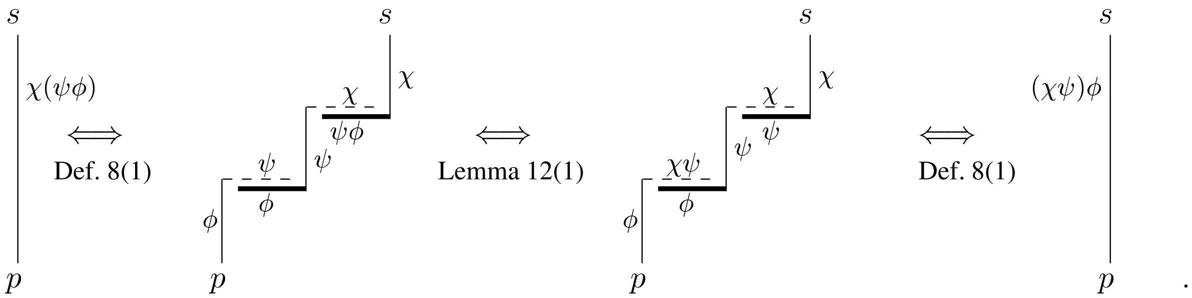


($\Longleftarrow$) Similar, but swap the roles of Prop. 11 and Def. 8(2). ■

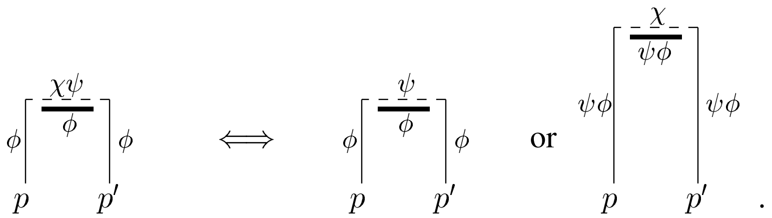
Proposition 13 *Composition of embeddings is associative.*

Proof Let $\phi : E_0 \to E_1$, $\psi : E_1 \to E_2$ and $\chi : E_2 \to E_3$. We require $\chi(\psi\phi) = (\chi\psi)\phi$.

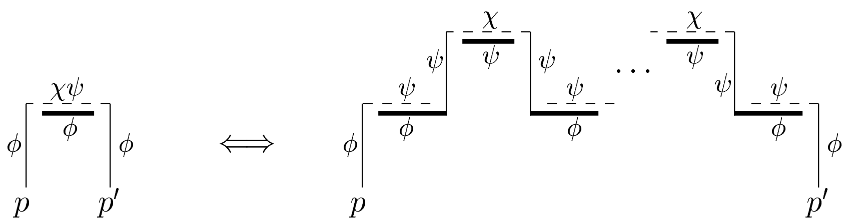
Consider the embedding maps first. All that is needed is to unpack Def. 8(1) four times, and then apply Lemma 12.



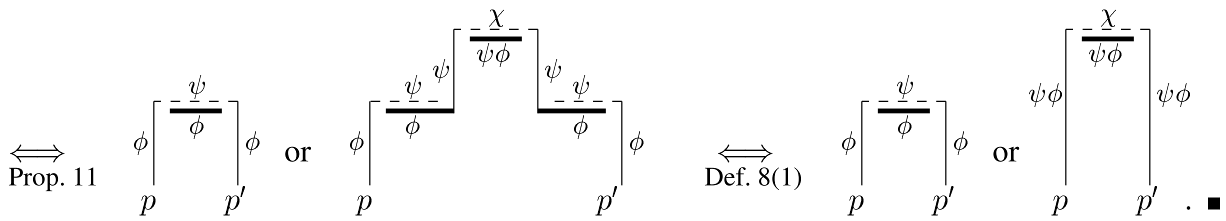
We now wish to show that $((\chi\psi)\phi)^{\mathsf{R}} = (\chi(\psi\phi))^{\mathsf{R}}$. Unpacking Def. 8(2) three times – for $(\chi\psi)\phi$, $\chi(\psi\phi)$ and $\psi\phi$ – we find that it will be sufficient to prove



Now by Def. 8(2) each $\chi\psi$ reflexion is either a ψ reflexion or a $\sqcap$-configuration, so



and, splitting between the cases of no $\sqcap$ and of at least one $\sqcap$, we complete the proof:



We are now ready to define our category of edgings and embeddings.

Definition 14 (category of edgings and embeddings) The category **E-Emb** has edgings as objects and edge embeddings as arrows, with composition as defined in Def. 8. Its *identity* embedding on each edging $E = (P, edg)$ is $\mathsf{id}_E : E \to E \stackrel{\text{def}}{=} (\mathsf{Id}_P, \emptyset)$. ∎

Theorem 15 **E-Emb** *is a category.*

Proof The associativity of composition is proved in Prop. 13, and it is easy to check that $\phi = \phi \circ \mathsf{id}_E = \mathsf{id}_F \circ \phi$ for any embedding $\phi : E \to F$.

4 Directed edgings and graphs

We now define the categories $\mathbf{E\text{-}Emb}_{\mathsf{d}}$ of directed edgings and $\mathbf{G\text{-}Emb}_{\mathsf{sd}}$ of shallow directed graphs, each with appropriate embeddings.

Definition 16 (directed edging) A *directed edging* is a triple (P, edg, δ), where (P, edg) is an edging and $\delta = (S, T)$ is a partition of P into *sources* S and *targets* T so that each edge in edg contains exactly one source and one target. ∎

Definition 17 (category of directed edge embeddings) Let (E_i, δ_i) be two directed edgings ($i = 0, 1$), with $\delta_i = (S_i, T_i)$. A *directed edge embedding* $\phi : (E_0, \delta_0) \to (E_1, \delta_1)$ is an edge embedding $\phi : E_0 \to E_1$ which *respects direction*, i.e. for all $p \in P_0$,

$$\begin{array}{ll} \text{If } p \in \mathsf{dom}\,\phi^{\mathsf{P}} \text{ then} & p \in S_0 \Leftrightarrow \phi^{\mathsf{P}} p \in S_1; \\ \text{If } p \in \mathsf{dom}\,\phi^{\mathsf{R}} \text{ then} & p \in S_0 \Leftrightarrow \phi^{\mathsf{R}} p \in T_0. \end{array}$$

These embeddings of directed edgings, with the composition and identities of **E-Emb**, constitute the category $\mathbf{E\text{-}Emb}_{\mathsf{d}}$ of directed edge embeddings. It is routine that the identities respect direction, and that composition preserves respect for direction. ∎

(The latter assertion relies on the fact that if ϕ respects direction then sources and targets alternate in a ϕ-reflexion chain.)

These definitions can now be easily lifted to the level of graphs. We define the categories $\mathbf{E\text{-}Emb}_{\mathsf{d}}$ of directed edgings and $\mathbf{G\text{-}Emb}_{\mathsf{sd}}$ of directed graphs, each with appropriate embeddings.

Definition 18 (directed graph) A *directed action graph* is a pair (G, δ), where $G = (L, U, ar, edg)$ is an action graph with port set P and (P, edg, δ) a directed edging. ■

Note that although δ partitions the ports of G equally, there is no further constraint on how it partitions the external link ports L. The reader may like to verify that the directed graphs G' and H' in Section 2 arise from G and H by imposing directions γ and δ, and that ϕ' thus arises from ϕ as detailed in Def. 17.

Definition 19 (category of directed graphs and embeddings) Let (G, γ) and (H, δ) be two directed graphs, with edgings (P, edg_G) and (Q, edg_H) respectively. Then a *directed graph embedding* $\phi : (G, \gamma) \to (H, \delta)$ is a graph embedding $\phi = (\phi^{\mathsf{V}}, \phi^{\mathsf{E}}) : G \to H$ such that $\phi^{\mathsf{E}} : (P, edg_G, \gamma) \to (Q, edg_H, \delta)$ is also a directed edge embedding.

These embeddings of directed graphs, with the composition and identities of $\mathbf{G\text{-}Emb}_{\mathsf{s}}$, constitute the category $\mathbf{G\text{-}Emb}_{\mathsf{sd}}$ of directed graph embeddings. The identities and composition clearly have the required properties. ■

5 Conclusion

To set up this category of graphs and embeddings has required an indirect approach. This is largely due to our firm intention to handle reflexion in full generality, and to admit graphs which contain link edges, because we expect these ideas to be essential in applications. Reflexion has great expressive power, e.g. it leads to succinct presentations of recursion, as Mifsud [13] has pointed out; it also eases the derivation of labelled transition systems [12] for action calculi in a way which guarantees congruential behavioural equivalences. We argued the importance of link edges in Section 2; we should point out further that they appear to be essential in modelling the important idea of *name fusion* originated by Parrow and Victor in the fusion calculus [15] and further developed by Gardner and Wischik as explicit fusions [7].

As we said in the introduction, this is the first of a sequence of papers which will develop the theory of linear and non-linear action graphs. The next paper, following the pattern of [5] but simpler (due to linearity) will set up the relationship between the category $\mathbf{G\text{-}Emb}_{\mathsf{s}}$ and the category $\mathbf{P\text{-}Cxt}_{\mathsf{s}}$ whose arrows are graphs and contexts; this will be shown to yield a "nice" reactive system in the sense of the Introduction, i.e. one for which we can uniformly derive behavioural congruences. In three further papers we intend to tackle (1) *nested* graphs; (2) quotienting our categories to yield *non-linear* graphs; (3) handling *multiple* graph embeddings and (correspondingly) *multihole* contexts, following Sewell [16]. This brings us close to real-life calculi; we may then begin to compare our general theory with their specific theories.

Acknowledgement We owe much to the insights of our colleagues Luca Cattani, Philippa Gardner and Peter Sewell.

References

[1] Abadi, M. and Gordon, A.D. (1997), A calculus for cryptographic protocols: the spi calculus. Proc. 4th ACM Conference on Computer and Communications Security, ACM Press, 36-47.

[2] Burstall, R.M. (1972), Some techniques for proving correctness of programs which alter data structures. In Machine Intelligence 7 (ed. Meltzer, Michie), Edinburgh University Press, 23–50.

[3] Cardelli, L. and Gordon, A.D. (2000), Mobile ambients. Foundations of System Specification and Computational Structures, LNCS 1378, 140–155.

[4] Cattani, G.L., private communication.

[5] Cattani, G.L., Leifer, J.J. and Milner, R. (2000),. Contexts and Embeddings for closed shallow action graphs. University of Cambridge Computer Laboratory, Technical Report 496. [Submitted for publication. Available at `http://www.cam.cl.ac.uk/users/jjl21` .]

[6] Gardner, P.A. and Wischik, L.G. (2000), Symmetric action calculi (abstract). University of Cambridge Computer Laboratory, in preparation.

[7] Gardner, P.A. and Wischik, L.G. (2000), Explicit fusions. Proc. MFCS 2000. LNCS 1893.

[8] Honda, K. (2000), Elementary structures in process theory (1): Sets with renaming. Mathematical Structures in Computer Science 10(5), 617–663.

[9] Joyal, A., Street, R. and Verity, D. (1996), Traced Monoidal Categories. Mathematical Proceedings of the Cambridge Philosophical Society, vol 119(3), pp 447–468.

[10] Lafont,Y. (1990), Interaction nets. Proc. 17th ACM Symposium on Principles of Programming Languages, 95–108.

[11] Leifer, J.J. (2001), Forthcoming PhD Dissertation, University of Cambridge.

[12] Leifer, J.J. and Milner, R. (2000), Deriving bisimulation congruences for reactive systems. Proc. CONCUR2000. [Available at `http://www.cam.cl.ac.uk/users/jjl21` .]

[13] Mifsud, A. (1995), Control structures. PhD Dissertation, Computer Science Department, University of Edinburgh.

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

[15] Parrow, J. and Victor, B. (1998), The fusion calculus: expressiveness and symmetry in mobile processes. Proc. LICS'98, IEEE Computer Society Press.

[16] Sewell, P. (1998), From rewrite rules to bisimulation congruences. Proc. CONCUR'98, LNCS 1466. [Revised version to appear in J. Theoretical Computer Science.]

[17] Turnbull, A., Forthcoming PhD Dissertation, University of Cambridge.