Technical Report
UCAM-CL-TR-581
ISSN 1476-2986
Number 581

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Axioms for bigraphical structure

Robin Milner

February 2004

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/TechReports/

Series editor: Markus Kuhn

ISSN 1476-2986

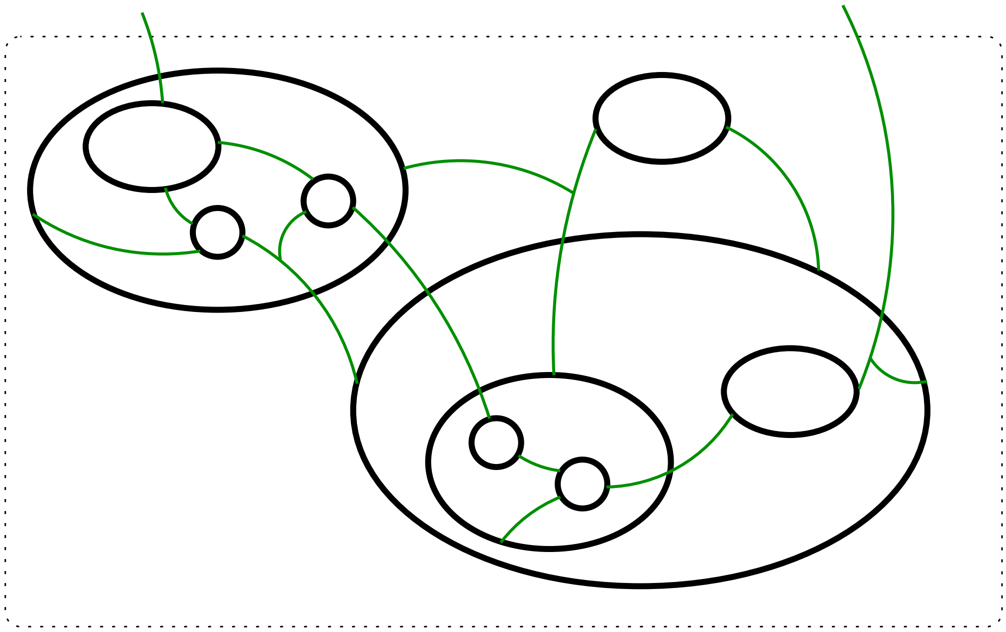
Axioms for bigraphical structure

Robin Milner

University of Cambridge, The Computer Laboratory,
J J Thomson Avenue, Cambridge CB3 0FD, UK

February 2004

Abstract. This paper axiomatises the structure of bigraphs, and proves that the resulting theory is complete. Bigraphs are graphs with double structure, representing locality and connectivity. They have been shown to represent dynamic theories for the π-calculus, mobile ambients and Petri nets, in a way that is faithful to each of those models of discrete behaviour. While the main purpose of bigraphs is to understand mobile systems, a prerequisite for this understanding is a well-behaved theory of the structure of states in such systems. The algebra of bigraph structure is surprisingly simple, as the paper demonstrates; this is because bigraphs treat locality and connectivity orthogonally.



1 Introduction

The diagram shows a bigraph, suppressing some of its detail. The ovals and circles are *nodes* which may be nested, and each node has *ports* which may be linked. The links are a partition of all the ports in a bigraph. Some links are external, shown here with wires escaping from the top of the diagram. When this bigraph is inserted in another (insertion will be represented by categorical composition) it will be placed in some region of that host graph, and each external link joined to some link of the host in a way that does not depend on the placing. Thus the independence of the linking and the placing of nodes, already illustrated by the way links cross boundaries in the diagram, is also respected by the operation of composing bigraphs.

The bigraphical model [16, 9] arises from a long-term effort beginning with action calculi [14, 6] to provide a theory common to different process calculi, and to base this theory on the topographical ideas that appear to pervade these calculi. These ideas are especially evident in the calculus of mobile ambients [1]; less obviously, they have also been found to inform the π-calculus [18]. A contributory effort [21, 2, 13, 12, 15] has been to unify the treatment of labelled transition systems by treating the labels as contexts, especially graphical contexts. Recently this unification has recovered existing behavioural theories for the π-calculus [8] and for mobile ambients [7], and has contributed to that for Petri nets [17].

In the categorical treatment of transition labels as contexts, the definition of a transition $a \xrightarrow{L} a'$ of an agent a specifies, among other things, that the composition $L \circ a$ (a in context L) performs a reaction; in other words, a and L may collaborate to perform it. This immediately suggests a categorical formulation in which the arrows are bigraphs, and the objects are interfaces explaining what kind of 'hole' in L will be occupied by the agent a, and what links will connect a to L. Considering static structure alone, this recalls Lawvere's categorical treatment of algebraic theories [11], in which the objects are simply finite ordinals and the arrows are tuples of terms; composition then is substitution of terms for variables in other terms. In bigraphs, term substitution is replaced by a more ramified notion of graph substitution.

The topic of this paper is to axiomatise the resulting structure of bigraphs. The justification for such a specific topic is threefold. First, the work already cited gives ample evidence that a graphical structure combining topography with connectivity has wide application in computer science; for as we have seen it brings unity to at least three models of discrete dynamics, each of which has already many applications. Second, it appears that the algebraic treatment of such dual structures has not been previously addressed; yet the behaviour of systems whose connectivity and topography are both reconfigurable may be so complex that their dynamics cannot be properly understood without a complete and rigorous treatment of their statics. Bigraphs are just one possible treatment of such dual structure, but it is likely that their static theory can be modified for other treatments. Third, as we shall see, dual structures seem to require a novel kind of normal form which is essential to a proof of axiomatic completeness.

We begin in Section 2 with an example, which illustrates the dynamic subtlety of a quite simple real-life system with mobile structure. In Section 3 we set up bigraphs as a category, and in Section 4 we explore their algebra enough to be able to propose the notion of *discrete normal form* (DNF). Section 5 introduces axioms, based upon those of a strict symmetric monoidal category, and proves their completeness using DNF. Section 6 introduces an alternative, *connected normal form* (CNF), that appears closer to process calculi and to programming languages, but uses a form of product that lends itself less well to axiomatisation. Finally, the concluding section mentions some related work and open problems.

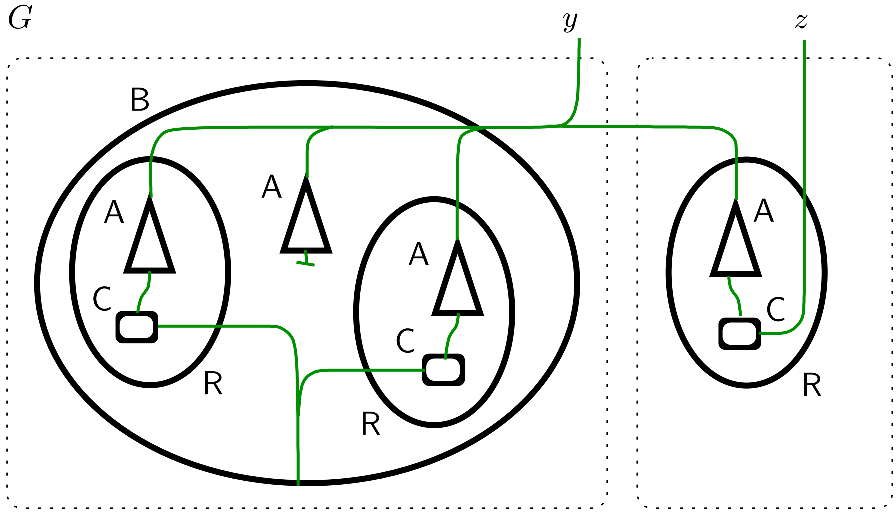


Figure 1: A bigraph G showing agents interacting in a structured environment

2 An example of bigraphs

The bigraph G in Figure 1 is a snapshot of part of a system in which people and things are interrelated and interacting. A bigraph consists principally of *nodes*, shown here with bold outlines. The nodes may be nested, and they have *ports* which may be connected by *links*. Each link many connect many ports; for example, all the A-nodes are joined by a single link represented by a forking bent line. The term 'bigraph' connotes this double structure of nesting and linking. The nesting of nodes imposes no constraint upon the linkage of their ports.

Every node is assigned a *control* (A, B, ...) which represents what kind of node it is and determines its *arity*, i.e. how many ports it has. In our example, the arities of A, B, C and R are 2, 1, 2 and 0 respectively. Here A-nodes represent agents, people equipped with mobile phones; the link among them indicates that they are conducting a conference call. If an agent is in a room (R) it may also be plugged in to a computer C. The computers in all rooms of a building (B) may be connected by a network, part of the building's infrastructure.

Our bigraph represents only part of a total (host) system. Some links have names; these are the *open* links that may be connected to other parts of the host. On the other hand the *closed* links, such as those joining three agents to their computers or two computers to the building, cannot be connected to other parts.

Every bigraph has a *width*; this one has width 2, shown by the two dotted rectangles, called *regions*. Nodes in the same region cannot be located within different nodes of the host; on the other hand, different regions may be arbitrarily far apart in the host.

As in all computing structures, some aspects of the behaviour of a subsystem can be analysed independently of its position within, or linkage to, its host system. In this case, we can imagine analysing the exchange of information among the four agents, and how it relates to their interaction with their computers, even though we know neither what building is occupied by the fourth agent, nor what other agents in the host are also

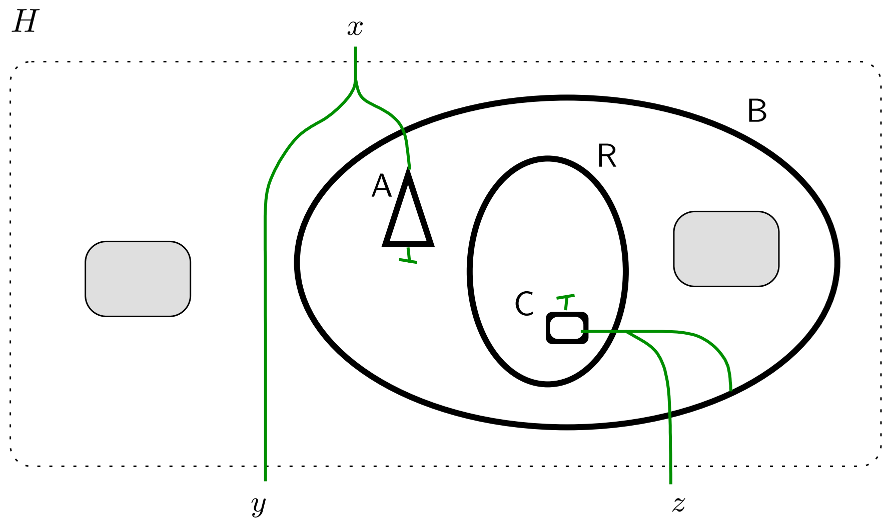


Figure 2: A host context H

taking part in the conference call (via the open link y).

Figure 2 illustrates a host bigraph H, which G may inhabit. H is a *contextual* bigraph, in two ways. It has two *holes*, the grey squares, which may be inhabited by a bigraph with two regions (such as G). It also has two *inner names*, y and z, for linking to an inhabitant bigraph via its corresponding *outer names*. We draw a bigraph's inner names below it, and its outer names above it. An inner name cannot be associated with a particular hole, because the corresponding link in an inhabitant belongs to no particular one of its regions; a link has no location (though each of its ports does).

Figure 3 shows the result $H \circ G$ of inserting G into the context H. We now know more about the conference call; it is being conducted by three people in one of the two buildings and two in the other. But we do not know what further context $H \circ G$ may inhabit; via its open link x the conference call could involve inhabitants of further buildings. It may not be obvious at first that the bigraph in Figure 3 represents the insertion of G into H (to see it, *first* place each region of G in the proper hole of H and *then* join homonymous inner and outer names); the reason is that placing and linking are in a sense orthogonal. This will be reflected in our mathematical formulation.

Before leaving our example, let us use it to illustrate the distinction between the structure (statics) and the activity (dynamics) of a system. Hitherto we have discussed only the structure of our example, and this paper addresses only the statics of bigraphs. But they admit a dynamical theory too, involving ways in which bigraphs may reconfigure their own placing and linking. In our example an agent may leave the conference call by severing itself from the link x, or may initiate further calls. Also, an agent may enter a room while talking; this could result in its connection to the computer installed in that room — a kind of dynamic binding. The dynamical theory of bigraphs has been somewhat developed [9], and shown to model behaviour in the π-calculus [8], mobile ambients [7] and Petri nets [17], in each case making a theoretical contribution to those disciplines.

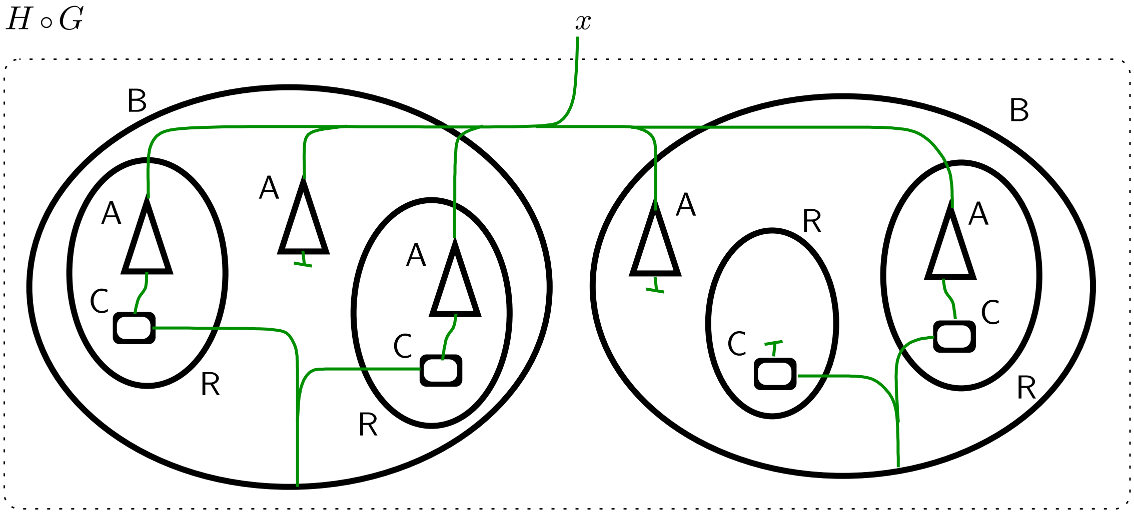


Figure 3: $H \circ G$, the partial system G embedded in host context H

3 Definitions

In this section we define the notion of *bigraph* formally, in terms of the constituent notions of *place graph* and *link graph*. We shall organise bigraphs as the arrows of a partial strict symmetric monoidal category whose objects are a simple form of interface. (We explain in a footnote immediately below what 'partial' means.) To prepare for the demonstration of a complete algebraic theory in the following section, we shall show how every bigraph can be expressed in a normal form that is unique up to isomorphism, in terms of elementary bigraphs.

Notation We use '$\circ$', 'id' and '$\otimes$' for composition, identity and tensor product in a category. Id_S denotes the identity function on a set S, and $\emptyset_S$ the empty function from $\emptyset$ to S. We use $S \uplus T$ for union of sets S and T known or assumed to be disjoint, and $f \uplus g$ for union of functions with domains known or assumed to be disjoint. We often interpret a natural number m as a finite ordinal $m = \{0, 1, \ldots, m-1\}$. We denote by $\vec{x}$ a finite sequence $\{x_i \mid i \in m\}$. We presuppose a denumerable set $\mathcal{X}$ of *names*.

Definition 3.1 (signature) A *signature* $\mathcal{K}$ is a set whose elements are called *controls*. For each control $\mathsf{K} \in \mathcal{K}$ it provides a finite ordinal $ar(\mathsf{K})$, an *arity*. ■

In refinements of the theory a signature may carry further information, such as a *sign* and/or a *type* for each port. The sign may be used, for example, to enforce the restriction that each negative port is connected to exactly one positive port, as in action calculi [2, 14]. Another possible refinement is a *kind* assigned to each node, determining the controls of the nodes it may contain. (An extreme case would be *atomic* nodes, which may contain no other nodes at all.)

Definition 3.2 (interface) An *interface* $I = \langle m, X \rangle$ consists of a finite ordinal m called a *width*, a finite set $X \subset \mathcal{X}$ called a *name set*.[1] ■

We are now ready to define bigraphs. We shall first define a *concrete* bigraph to be the combination of two constituents, a place graph and a link graph.

Definition 3.3 (concrete bigraph) A *concrete bigraph* over the signature $\mathcal{K}$ takes the form $G = (V, E, ctrl, G^{\mathsf{P}}, G^{\mathsf{L}}) : I \to J$ where the interfaces $I = \langle m, X \rangle$ and $J = \langle n, Y \rangle$ are its *inner* and *outer faces*. Its first two components V and E are finite sets of *nodes* and *edges* respectively. The third component $ctrl : V \to \mathcal{K}$, a *control map*, assigns a control to each node. The remaining two are:

$$\begin{array}{ll} G^{\mathsf{P}} = (V, ctrl, prnt) : m \to n & \text{a } \textit{place graph} \\ G^{\mathsf{L}} = (V, E, ctrl, link) : X \to Y & \text{a } \textit{link graph}\ . \end{array}$$ ■

We shall define the composition, identities and tensor product of bigraphs in terms of these operations on their constituents. Let us take place graphs first.

Definition 3.4 (place graph) A *place graph* $G = (V, ctrl, prnt) : m \to n$ has an *inner width* m and an *outer width* n, both finite ordinals; a finite set V of nodes with a control map $ctrl : V \to \mathcal{K}$; and a *parent map* $prnt : m \uplus V \to V \uplus n$. The parent map is *acyclic*, i.e. $prnt^k(v) \neq v$ for all $k > 0$ and $v \in V$. ■

The acyclicity condition makes the parent map $prnt$ represent a forest of n unordered trees. The widths m and n of $G : m \to n$ index its *sites* $0, \ldots, m$–1 and *roots* $0, \ldots, n$–1 respectively. The sites and nodes —i.e. the domain of $prnt$— are called *places*. Two places are *siblings* if they have the same parent. A node or root is *barren* if it has no children.

The sites and roots provide the means of composing the forests of two place graphs; each root i of the first is planted in site i of the second. Figure 4 shows a simple example of composing place graphs; note the correspondence between the sites of H and the roots of G. Formally, let $G_i = (V_i, ctrl_i, prnt_i) : m_i \to m_{i+1}$ $(i = 0, 1)$ be place graphs with $V_0 \cap V_1 = \emptyset$. Then $G_1 \circ G_0 \stackrel{\text{def}}{=} (V, ctrl, prnt)$ where $V = V_0 \uplus V_1$, $ctrl = ctrl_0 \uplus ctrl_1$, and

$$prnt = (\mathsf{Id}_{V_0} \uplus prnt_1) \circ (prnt_0 \uplus \mathsf{Id}_{V_1})\ .$$

The identity place graph at m is $\mathsf{id}_m \stackrel{\text{def}}{=} (\emptyset, \emptyset_{\mathcal{K}}, \mathsf{Id}_m) : m \to m$.

The tensor product of two place graphs $G : k \to \ell$ and $H : m \to n$ with disjoint node sets is $G \otimes H : k{+}m \to \ell{+}n$. It consists simply of placing the two forests side-by-side, and we need not define it more formally.

We turn now to link graphs.

[1] We saw informally in the previous section that names are used to link bigraphs together. A more abstract presentation would dispense with alphabetic names entirely, representing them positionally — as in abstract treatments of the λ-calculus. This is perfectly possible here. Indeed, the word ‘partial’ used above would then be redundant; it refers to the fact that the tensor product of interfaces is defined only when their name sets are disjoint, and that condition is unnecessary with positional notation. We prefer alphabetic names here; it makes little difference to the mathematics, and allows a much more lucid connection to be made to process calculi.

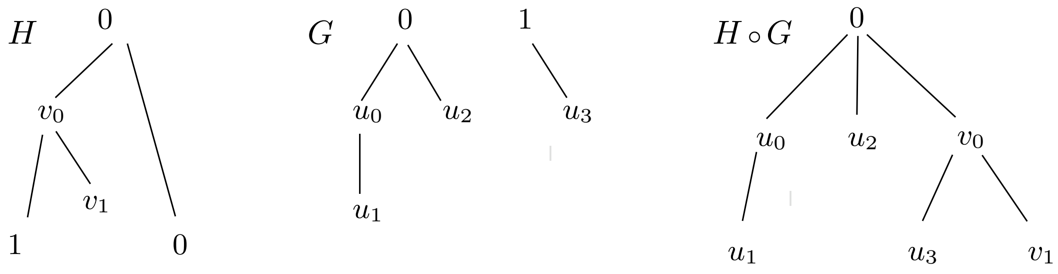


Figure 4: Composing two place graphs

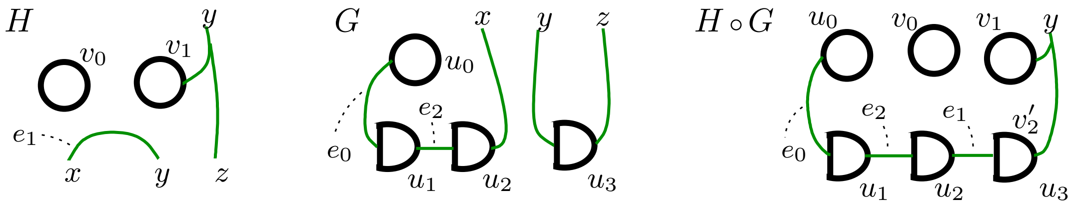


Figure 5: Composing two link graphs

Definition 3.5 (link graph) A *link graph* $G = (V, E, ctrl, link) : X \to Y$ has finite sets X of *inner names*, Y of *outer names*, V of *nodes* and E of *edges*. It also has a function $ctrl : V \to \mathcal{K}$ called the *control map*, and a function $link : X \uplus P \to E \uplus Y$ called the *link map*, where the disjoint sum $P \stackrel{\text{def}}{=} \sum_{v \in V} ar(ctrl(v))$ is the set of *ports* of G. ■

The inner names X and ports P are the *points* of G, and the edges E and outer names Y its *links*. A link is *idle* if it has no preimage under the $link$ map. A link graph is *lean* if it has no idle edges. An outer name is an *open* link, an edge is a *closed* link. A point (i.e. an inner name or port) is *open* if its link is open, otherwise *closed*. Two distinct points are *peers* if they are in the same link.

It may seem superfluous to admit the possibility of an idle link, especially one that is open, because it represents an outer name y that 'names' nothing. But the dynamic rules of a bigraphical system allow such a name to arise. In the π-calculus it arises when a communication on the channel y has occurred and no further use of this channel remains. A similar situation can give rise to a barren root in a place graph.

Figure 5 shows a simple example of composing link graphs. Formally, let $G_i = (V_i, E_i, ctrl_i, link_i) : X_i \to X_{i+1}$ $(i = 0, 1)$ be two link graphs with $V_0 \cap V_1 = E_0 \cap E_1 = \emptyset$. Then $G_1 \circ G_0 \stackrel{\text{def}}{=} (V, E, ctrl, link)$ where $V = V_0 \uplus V_1$, $ctrl = ctrl_0 \uplus ctrl_1$, $E = E_0 \uplus E_1$ and

$$link = (\mathsf{Id}_{E_0} \uplus link_1) \circ (link_0 \uplus \mathsf{Id}_{P_1}) .$$

We can describe the composite link map $link$ of $G_1 \circ G_0$ as follows, considering all possible arguments $p \in X_0 \uplus P_0 \uplus P_1$:

$$link(p) = \begin{cases} link_0(p) & \text{if } p \in X_0 \uplus P_0 \text{ and } link_0(p) \in E_0 \\ link_1(x) & \text{if } p \in X_0 \uplus P_0 \text{ and } link_0(p) = x \in X_1 \\ link_1(p) & \text{if } p \in P_1 \ . \end{cases}$$

The identity link graph at X is $\mathsf{id}_X \stackrel{\text{def}}{=} (\emptyset, \emptyset, \emptyset_{\mathcal{K}}, \mathsf{Id}_X) : X \to X$.

The tensor product of two link graphs $G\colon W \to X$ and $H\colon Y \to Z$ can be formed provided that their node sets and edge sets are disjoint and that $W \cap Y = X \cap Z = \emptyset$. It is $G \otimes H\colon W \uplus Y \to X \uplus Z$, and consists simply of the union of their link maps.

We are now ready to define the category that is the main object of study in this paper. Recall that in Definition 3.3 a concrete bigraph $G\colon \langle m, X\rangle \to \langle n, Y\rangle$ consists of a combination of a place graph $G^{\mathsf{P}}: m \to n$ and a link graph $G^{\mathsf{L}}: X \to Y$ having the same node set and control map. We shall write such a combination as $G = \langle G^{\mathsf{P}}, G^{\mathsf{L}}\rangle$.

Definition 3.6 (monoidal category of bigraphs) The composition of two concrete bigraphs $G = \langle G^{\mathsf{P}}, G^{\mathsf{L}}\rangle\colon I \to J$ and $H = \langle H^{\mathsf{P}}, H^{\mathsf{L}}\rangle\colon J \to K$ with disjoint node sets and disjoint edge sets is

$$H \circ G \stackrel{\text{def}}{=} \langle H^{\mathsf{P}} \circ G^{\mathsf{P}}, H^{\mathsf{L}} \circ G^{\mathsf{L}}\rangle\colon I \to K \ .$$

Two concrete bigraphs G_0 and G_1 are said to be *lean-support equivalent*, $G_0 \Bumpeq G_1$, if they differ only by a bijection between their nodes and between their non-idle edges; idle edges are ignored. An *abstract bigraph* consists of a $\Bumpeq$-equivalence class of concrete bigraphs. Composition and identity of abstract bigraphs are given by

$$\begin{aligned} [H]_{\Bumpeq} \circ [G]_{\Bumpeq} &\stackrel{\text{def}}{=} [\langle H^{\mathsf{P}} \circ G^{\mathsf{P}}, H^{\mathsf{L}} \circ G^{\mathsf{L}}\rangle]_{\Bumpeq} \\ \mathsf{id}_{\langle m,X\rangle} &\stackrel{\text{def}}{=} [\langle \mathsf{id}_m, \mathsf{id}_X\rangle]_{\Bumpeq} \ . \end{aligned}$$

The tensor product of two interfaces with disjoint name sets is

$$\langle m, X\rangle \otimes \langle n, Y\rangle \stackrel{\text{def}}{=} \langle m+n, X \uplus Y\rangle \ .$$

The tensor product of two abstract bigraphs $F\colon H \to I$ and $G\colon J \to K$, where $H \otimes J$ and $I \otimes K$ are defined, is given by

$$[F]_{\Bumpeq} \otimes [G]_{\Bumpeq} \stackrel{\text{def}}{=} [\langle F^{\mathsf{P}} \otimes G^{\mathsf{P}}, F^{\mathsf{L}} \otimes G^{\mathsf{L}}\rangle]_{\Bumpeq}\colon H \otimes J \to I \otimes K \ . \qquad \blacksquare$$

Figure 6 shows the composition of two bigraphs; they are the combinations of the place graphs and link graphs composed in Figures 4 and 5. The labelling of sites in H, indicates where each root of a client bigraph (such as G) should be planted. For clarity, nodes are identified in the figure; in an abstract bigraph these identifiers are forgotten.

The reader may wonder whether the algebra of bigraphs can be factored into two separate algebras, one for placing and the other for linking. But in each of these separate algebras the identity of nodes would have been forgotten (just as in ordinary algebra the identity of subterms is forgotten when one term is substituted in another), and we have seen that the combination of a place graph with a link graph to form a bigraph depends crucially on the identity of nodes. Much of the subtlety of bigraph algebra stems from how this combination interacts with the algebraic operations of composition and tensor product.

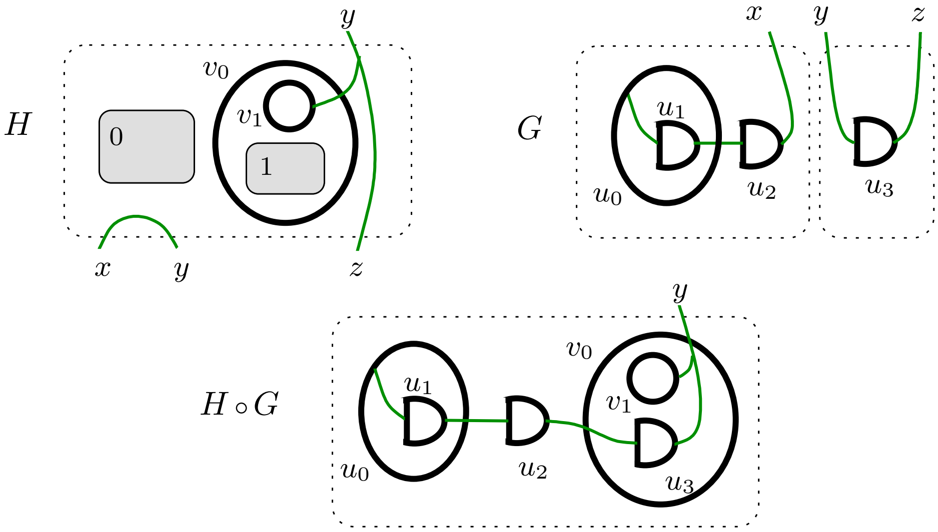


Figure 6: Composing two bigraphs

4 Algebra and discrete normal form

In this section we give the elementary bigraphs from which all others can be formed by composition and tensor product. We then define a normal form for bigraphical expressions, called *discrete normal form* (DNF), and show how each bigraph can be expressed in DNF uniquely (up to isomorphism).

To avoid too many parentheses in expressions we shall often represent composition by juxtaposition; it binds tightly, for example $G_1 G_2 \otimes G_3$ means $(G_1 \circ G_2) \otimes G_3$.

There are two degenerate forms of interface: a *place interface* $\langle m, \emptyset \rangle$, and a *link interface* $\langle 0, X \rangle$. We write them as m and X (or just x if $X = \{x\}$) respectively. The fully degenerate form is the *origin* $\epsilon \stackrel{\text{def}}{=} \langle 0, \emptyset \rangle$, which is of course the unit for tensor product on interfaces. An important class of bigraphs are the *ground bigraphs* $G \colon \epsilon \to I$, those whose inner face is the origin. If G is ground then it has no holes or inner names; indeed there is no useful composition GF, since (by the properties of a strict monoidal category) it can be written as a product $G \otimes F$.

A *placing* is a bigraph $m \to n$ with no nodes. All placings can be expressed in terms of three kinds:

$$\begin{array}{rcll} 1 & : & \epsilon \to 1 & \text{a barren root} \\ \mathit{merge} & : & 2 \to 1 & \text{map two sites to one root} \\ \gamma_{m,n} & : & m+n \to n+m & \text{swap } m \text{ with } n \text{ places.} \end{array}$$

We use π, ρ to range over *permutations*, those placings generated by composition and tensor product from the $\gamma_{m,n}$. For all $m \geq 0$ we can define $\mathit{merge}_m \colon m \to 1$, which merges m sites, as follows:

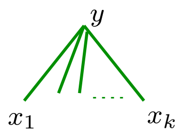
$$\begin{aligned} \mathit{merge}_0 &\stackrel{\text{def}}{=} 1 \\ \mathit{merge}_{m+1} &\stackrel{\text{def}}{=} \mathit{merge}(\mathsf{id}_1 \otimes \mathit{merge}_m) \,. \end{aligned}$$

Note that $merge_1 = \mathsf{id}_1$, and hence $merge_2 = merge$.

A *linking* or *wiring* is a bigraph $X \to Y$, which necessarily has no nodes. All linkings can be expressed in terms of two kinds:

$$\begin{array}{rcll} /x & : & x \to \epsilon & \text{closure} \\ {}^{y}/X & : & X \to y & \text{substitution } x \mapsto y \text{ (all } x \in X) \,. \end{array}$$

a closure $/x$

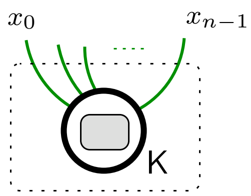


a substitution ${}^{y}/X$

A closure just closes a single link. For $X = \{x_1, \ldots, x_k\}$ we define the multiple closure $/X \stackrel{\text{def}}{=} /x_1 \otimes \cdots \otimes /x_k$. For $X = X_1 \uplus \cdots \uplus X_n$ and $Y = \{y_1, \ldots, y_n\}$, a multiple substitution $\sigma\colon X \to Y$ is defined by ${}^{y_1}/X_1 \otimes \cdots \otimes {}^{y_n}/X_n$. A substitution need not be surjective. We write $Y\colon \epsilon \to Y$ for the empty substitution, or just $y\colon \epsilon \to y$ if $Y = \{y\}$; these are the duals of closures. We shall use ω to range over linkings, σ, τ over substitutions, and α, β over the bijective substitutions, which we call *renamings*.

Permutations π and renamings α together generate all isomorphisms in the category of bigraphs; in fact every isomorphism takes the form $\pi \otimes \alpha$.

The only other elementary bigraph is a *discrete ion* $\mathsf{K}_{\vec{x}}\colon 1 \to \langle 1, \{\vec{x}\}\rangle$, for any sequence $\vec{x} = x_1, \ldots, x_k$ of distinct names where $k = ar(\mathsf{K})$.



a discrete ion $\mathsf{K}_{\vec{x}}$

We now turn to the first of our normal forms. It depends on two important concepts:

Definition 4.1 (prime, discrete) An interface is *prime* if it has unit width. It takes the form $\langle 1, X\rangle$, which we shall often abbreviate to $\langle X\rangle$. A bigraph $G\colon I \to J$ is *prime* if J is prime and I has no names. A bigraph is *discrete* if every link is open and contains exactly one point. ■

Thus a discrete ion is an instance of a prime discrete bigraph. More generally we define a *discrete molecule* M to be a prime discrete bigraph having a single outermost node.

Figure 7 shows an example of a discrete bigraph. Note that is consists just of discrete ions in a topographical arrangement. It is in fact the discretisation of the bigraph G from Figure 5; by composing it with a linking we recover G.

We shall now express this insight in a general form. We identify four levels of structure in bigraphs, as four forms of expression. At each level, the expression is unique up to certain isomorphisms. Taken in reverse order, these forms represent the expression of any bigraph G in *discrete normal form* (DNF). The following proposition

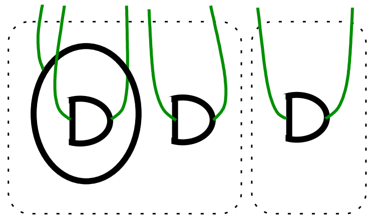


Figure 7: A discrete bigraph

treats P, D, α etc as *expressions* for bigraphs, but equality '$=$' is semantical; for example $P = P'$ means not that P and P' are identical expressions but that they denote the same bigraph. We omit a proof of the proposition; it follows routinely from the formal definitions in the preceding section.

Proposition 4.2 (discrete normal form)

1. *A discrete molecule M may be expressed as*

$$M = (\mathsf{K}_{\vec{x}} \otimes \mathsf{id}_Y)P$$

where P is prime and discrete. Any other such expression of M takes the form $(\mathsf{K}_{\vec{x}} \otimes \mathsf{id}_Y)P'$ where $P' = P$.

2. *A discrete prime P may be expressed as*

$$P = (\mathit{merge}_{n+k} \otimes \mathsf{id}_Y)(\mathsf{id}_n \otimes M_0 \otimes \cdots \otimes M_{k-1})\pi$$

where each M_i is a discrete molecule. Any other such expression of P takes the form $(\mathit{merge}_{n+k} \otimes \mathsf{id}_Y)(\mathsf{id}_n \otimes M'_0 \otimes \cdots \otimes M'_{k-1})\pi'$, where there exist ρ, ρ_i ($i \in k$) and ρ' such that $M'_i = M_{\rho(i)}\rho_i$ and $(\mathsf{id}_n \otimes \rho_0 \otimes \cdots \otimes \rho_{n-1})\pi' = (\mathsf{id}_n \otimes \rho')\pi$. Moreover if ℓ_i is the inner width of M_i for each $i \in k$ and $\vec{\ell} = \ell_0, \ldots, \ell_{k-1}$, then $\rho' = \overline{\rho}_{\vec{\ell}}$ as defined in Lemma 5.2.

3. *A discrete bigraph D with outer width n may be expressed as*

$$D = (P_0 \otimes \cdots \otimes P_{n-1})\pi \otimes \alpha$$

where each P_i is prime and discrete. Any other such expression of D takes the form $(P'_0 \otimes \cdots \otimes P'_{n-1})\pi' \otimes \alpha$ where $P'_i = P_i\rho_i$ and $(\rho_0 \otimes \cdots \otimes \rho_{n-1})\pi' = \pi$ for certain permutations ρ_i.

4. *A bigraph G with outer width n may be expressed as*

$$G = (\mathsf{id}_n \otimes \omega)D$$

where D is discrete. Any other such expression of G takes the form $(\mathsf{id}_n \otimes \omega')D'$ where $\omega' = \omega\alpha$ and $(\mathsf{id}_n \otimes \alpha)D' = D$ for some renaming α.

This proposition details the isomorphisms that allow variation of expression at each level. For example, consider a discrete bigraph $D\colon \langle m, X\rangle \to \langle n, Y\rangle$. Each inner name $x \in X$ is linked to a distinct name $y \in Y$, and has no peers; this fully determines the renaming α. There are no links between regions of a discrete bigraph, so D is indeed a product of primes. However, the order of sites within each prime P_i may be chosen arbitrarily, and the product needs to be composed with a permutation correspondingly chosen to yield the correct order among all the sites of D.

Unique factorisation of a discrete bigraph D into primes justifies our definition of 'prime'; unicity would fail without the constraint that a prime has no inner names.

5 Axioms and completeness

We now address the question: What set of axioms is complete for equations between bigraph expressions, in the sense that every valid equation is provable? Recall that we are considering expressions built by composition, identities and tensor product from the six classes of constants:

$$1 \qquad \mathit{merge} \qquad \gamma_{m,n} \qquad /x \qquad {}^{x}/Y \qquad \mathsf{K}_{\vec{x}}$$

where the $\vec{x}$ are distinct. The answer to our question turns out to be rather simple. First, we must extend the symmetries $\gamma_{m,n}$ to arbitrary interfaces by defining

$$\gamma_{I,J} \stackrel{\text{def}}{=} \gamma_{m,n} \otimes \mathsf{id}_{X \uplus Y} \quad \text{where } I = \langle m, X\rangle, J = \langle n, Y\rangle \ .$$

The axioms are shown in Table 1. The categorical axioms are standard for a strict symmetric monoidal category. But note that the tensor product is defined only when interfaces have disjoint name sets; thus the equations are required to hold only when both sides are defined. What is remarkable is that no axioms are required on ions except a simple renaming axiom (needed only because names are treated positionally). Thus bigraphs are a rather free structure.

In what follows we shall use $E = F$, instead of the more verbose $\vDash E = F$, to mean that the two expressions denote the same bigraph; when we are talking of equality inferred from the axioms we always write $\vdash E = F$.

The *soundness* of our axioms, i.e. that $\vdash E = F$ implies $E = F$, is obvious and no formal proof is needed. We shall say that the axiomatic theory is *complete* for bigraph expressions in a class $\mathcal{E}$ if $E = F$ implies $\vdash E = F$ for all $E, F \in \mathcal{E}$. We shall prove completeness for increasingly large classes $\mathcal{E}$, culminating in the full class. We often talk of a class *generated* by certain elementary expressions; this means the class formed from those elements by composition, identities and product.

Preliminaries

In this subsection we prove some useful lemmas, including completeness for *place expressions* and *link expressions*, those generated from placings and linkings respectively, and thence for ion-free expressions. We also introduce *linearity*, a syntactic property that matches discreteness.

CATEGORICAL AXIOMS:

$$
\begin{array}{rcl}
A\,\mathsf{id} = & A & = \mathsf{id}\,A \\
A(BC) & = & (AB)C \\
A\otimes \mathsf{id}_\epsilon = & A & = \mathsf{id}_\epsilon \otimes A \\
A\otimes(B\otimes C) & = & (A\otimes B)\otimes C \\
(A_1\otimes B_1)(A_0\otimes B_0) & = & (A_1A_0)\otimes(B_1B_0) \\
\gamma_{I,\epsilon} & = & \mathsf{id}_I \\
\gamma_{J,I}\gamma_{I,J} & = & \mathsf{id}_{I\otimes J} \\
\gamma_{I,K}(A\otimes B) & = & (B\otimes A)\gamma_{H,J} \qquad (A\colon H\to I, B\colon J\to K)
\end{array}
$$

LINK AXIOMS:

$$
\begin{array}{rcl}
/y\circ {}^y/x & = & /x \\
/y\circ y & = & \mathsf{id}_\epsilon \\
{}^z/(Y\uplus y)\circ(\mathsf{id}_Y\otimes {}^y/X) & = & {}^z/(Y\uplus X)
\end{array}
$$

PLACE AXIOMS:

$$
\begin{array}{rcll}
\mathit{merge}(1\otimes \mathsf{id}_1) & = & \mathsf{id}_1 & \text{(unit)} \\
\mathit{merge}(\mathit{merge}\otimes \mathsf{id}_1) & = & \mathit{merge}(\mathsf{id}_1\otimes \mathit{merge}) & \text{(associative)} \\
\mathit{merge}\gamma_{1,1} & = & \mathit{merge} & \text{(commutative)}
\end{array}
$$

NODE AXIOMS:

$$
(\mathsf{id}_1\otimes\alpha)\mathsf{K}_{\vec{x}} = \mathsf{K}_{\alpha(\vec{x})}\ .
$$

Table 1: Axioms for bigraph equality

Lemma 5.1

1. $\vdash \mathit{merge}_m \pi = \mathit{merge}_m$

2. $\vdash \mathit{merge}_k(\mathit{merge}_{m_0} \otimes \cdots \otimes \mathit{merge}_{m_{k-1}}) = \mathit{merge}_m$, *where* $m = \sum_i m_i$.

Since permutations are generated by the $\gamma_{m,n}$ the axiom $\gamma_{I,K}(A \otimes B) = (B \otimes A)\gamma_{H,J}$ can be iterated to push a permutation through any product of primes, as follows:

Lemma 5.2 *Let* $P_i : m_i \to \langle 1, X_i \rangle$ *be prime expressions for* $i \in n$, *with* $X = X_0 \uplus \cdots \uplus X_{n-1}$, *and let* π *be a permutation on* n. *Then there exists a permutation which we denote by* $\overline{\pi}_{\vec{m}}$, *dependent only on* π *and* $\vec{m}$, *such that*

$$\vdash (\pi \otimes \mathsf{id}_X)(P_0 \otimes \cdots \otimes P_{n-1}) = (P_{\pi(0)} \otimes \cdots \otimes P_{\pi(n-1)})\overline{\pi}_{\vec{m}} \ .$$

We are now ready to prove some special instances of completeness.

Lemma 5.3 *The theory is complete for place expressions.*

Proof First, it is standard in strict symmetric monoidal categories that the categorical axioms are complete for the permutation expressions π generated from the $\gamma_{m,n}$.

Next we show for every place expression E that

$$\vdash E = (\mathit{merge}_{m_0} \otimes \cdots \otimes \mathit{merge}_{m_{k-1}})\pi$$

for some $k \geq 0$ and permutation expression π. The proof is by structural induction on expressions. For the inductive step, if it holds for E and F then it immediately holds for $E \otimes F$, and to show it for EF amounts to a simple use of Lemmas 5.1 and 5.2.

Now suppose F is another place expression with $E = F$ and

$$\vdash F = (\mathit{merge}_{n_0} \otimes \cdots \otimes \mathit{merge}_{n_{\ell-1}})\pi'$$

Then we must have $k = \ell$, $m_i = n_i$, and $\pi' = (\rho_0 \otimes \cdots \otimes \rho_{k-1})\pi$ for some ρ_i. Then this equation is provable by completeness for permutation expressions. We also have $\vdash \mathit{merge}_{m_i}\rho_i = \mathit{merge}_{m_i}$ from Lemma 5.1, so we deduce $\vdash E = F$ as required. ∎

Lemma 5.4 *The theory is complete for link expressions.*

Proof First we show for every link expression E that $\vdash E = \widehat{E}$, where $\widehat{E}$ is a *link normal form* consisting of a product of zero or more terms of the three types

$$/y \qquad {}^{x}/Y \qquad /x \circ {}^{x}/Y$$

where in the second type Y may be any set, but in the third type Y must contain at least two members. (This type represents a closed link between two or more inner names.) Again the proof is by structural induction on E, and the only case needing careful attention is to show that if the property hold for E and F then it holds for EF. So consider $\widehat{E}\widehat{F}$, and consider the terms ${}^{y}/Z$ of the second type in $\widehat{F}$. Each term in $\widehat{E}$ composes with a subset of these terms, and each of these compositions can be proved by the axioms equal to a term that is again one of the three types. This yields the normal form $\widehat{EF}$. We leave details to the reader.

Since our axioms are sound, if $E = F$ then also $\widehat{E} = \widehat{F}$. But our normal form is such that two of them denoting the same bigraph must contain exactly the same terms, perhaps differently ordered. So $\vdash \widehat{E} = \widehat{F}$, hence $\vdash E = F$ as required. ■

Lemma 5.5 *The theory is complete for ion-free expressions.*

Proof By an easy structural induction on expressions we can prove that $\vdash E = E^{\mathsf{P}} \otimes E^{\mathsf{L}}$, the product of a place expression and a link expression. If $E = F$ we then deduce that $E^{\mathsf{P}} = F^{\mathsf{P}}$ and $E^{\mathsf{L}} = F^{\mathsf{L}}$, since E^{P} and F^{P} are essentially the place graphs of the two bigraphs; similarly with their link graphs. Hence $\vdash E^{\mathsf{P}} = F^{\mathsf{P}}$ and $\vdash E^{\mathsf{L}} = F^{\mathsf{L}}$ by Lemmas 5.3 and 5.4, and finally $\vdash E = F$ as required. ■

In order to reduce our completeness problem to one for discrete bigraphs we need a syntactic version of DNF. A discrete bigraph has many syntactic expressions; in particular, we shall show that it has a linear expression, defined as follows:

Definition 5.6 (linear) A bigraph expression is *linear* if it contains no substitution elements except *linear* ones, those of the form y/x. ■

Clearly all sub-expressions of a linear expression are linear; thus linearity is amenable to structural induction. For example:

Lemma 5.7 *If E is linear then $\vdash E = E' \otimes \alpha$ where E' is linear without inner names.*

Proof By structural induction. The proof for elements and the inductive step for product are easy. It remains to show that if the property holds for E and F then it holds for EF. Let EF be linear with $\vdash E = E' \otimes \alpha$ and $\vdash F = F' \otimes \beta$, where E' and F' have no inner names. We have $\vdash \alpha = \alpha_0 \otimes \alpha_1$, where the domains of α_0, α_1 are the outer names of F and β respectively. Then $\vdash EF = (E' \otimes \alpha_0)F' \otimes \alpha_1\beta$, which is of the required form. ■

We end this subsection with a first approximation to a provable normal form for arbitrary bigraph expressions:

Proposition 5.8 (underlying linear expression) *For any expression G of outer width m there exist ω and linear E such that $\vdash G = (\mathsf{id}_m \otimes \omega)E$.*

Proof Again by structural induction, and we need only look at the inductive step for composition. Consider GH, and assume that for some linear E and F

$$\vdash G = (\mathsf{id}_m \otimes \omega)E \quad \text{and} \quad \vdash H = (\mathsf{id}_n \otimes \omega')F \ .$$

Then by Lemma 5.7 we have $\vdash E = E' \otimes \alpha$ where E' has no inner names. Hence

$$\begin{aligned} \vdash GH &= (\mathsf{id}_m \otimes \omega)(E' \otimes \alpha\omega')F \\ &= (\mathsf{id}_m \otimes \omega(\mathsf{id}_Y \otimes \alpha\omega'))(E' \otimes \mathsf{id}_X)F \end{aligned}$$

where X and Y are the outer names of F and E' respectively. This is of the required form, since $(E' \otimes \mathsf{id}_X)F$ is linear. ■

Provable normal forms

We are interested in four increasingly general kinds of expression: those that denote respectively discrete molecules, discrete primes, discrete bigraphs and arbitrary bigraphs. We now set out to prove that every expression of each kind can be proved equal to the corresponding kind of DNF, defined as follows (as suggested in the previous section):

Definition 5.9 (discrete normal forms) There are four kinds of DNF: MDNFs M for discrete molecules, PDNFs P for discrete primes, DDNFs D for discrete bigraphs and BDNFs B for bigraphs:

$$\begin{array}{rl} \text{MDNF:} & M ::= (\mathsf{K}_{\vec{x}} \otimes \mathsf{id}_Y)P \\ \text{PDNF:} & P ::= (\mathit{merge}_{n+k} \otimes \mathsf{id}_Y)(\mathsf{id}_n \otimes M_0 \otimes \cdots \otimes M_{k-1})\pi \\ \text{DDNF:} & D ::= (P_0 \otimes \cdots \otimes P_{n-1})\pi \otimes \alpha \\ \text{BDNF:} & B ::= (\mathsf{id}_n \otimes \omega)D\ . \end{array}$$

■

We begin with a lemma showing that normal forms are provably closed under isomorphism, in a certain sense:

Lemma 5.10 *Let $B\colon I \to I'$ be a* BDNF. *If ι and ι' are isomorphisms on I and I' respectively, then $\vdash \iota' B \iota = B'$ for some* BDNF *B'.*

The analogous property holds also for DDNF, PDNF *and* MDNF.

We omit the proof, which depends upon Lemma 5.2; it uses induction on the number of ions in an expression. Our next lemma uses a more complex instance of this technique, and shows that DNFs are provably closed under certain compositions:

Lemma 5.11 *Let $C\colon \ell \to \langle m, Z\rangle$ be a product of* PDNFs. *Then*

1. *If $M\colon m \to \langle Y\rangle$ is a* MDNF *then $\vdash (M \otimes \mathsf{id}_Z)C = M'$ for some* MDNF *M'.*
2. *If $P\colon m \to \langle Y\rangle$ is a* PDNF *then $\vdash (P \otimes \mathsf{id}_Z)C = P'$ for some* PDNF *P'.*
3. *If $D\colon m \to \langle n, Y\rangle$ is a* DDNF *then $\vdash DC = D'$ for some* DDNF *D'.*

Proof We first prove (1) and (2) by simultaneous induction on the number n of ions in P or M. Assume both parts hold for $< n$ ions.

For (1) with n ions, let M be $(\mathsf{K}_{\vec{x}} \otimes \mathsf{id}_W)P$ for PDNF $P\colon m \to \langle W\rangle$. Then

$$\vdash (M \otimes \mathsf{id}_Z)C = (\mathsf{K}_{\vec{x}} \otimes \mathsf{id}_{W \uplus Z})(P \otimes \mathsf{id}_Z)C$$

where P has $n-1$ ions. We may therefore apply (2) to P, yielding the required result.

For (2) with n ions, let P be $(\mathit{merge}_{h+k} \otimes \mathsf{id}_Y)(\mathsf{id}_h \otimes M_0 \otimes \cdots \otimes M_{k-1})\pi$. Now C is a product $Q_0 \otimes \cdots \otimes Q_{m-1}$ of PDNFs, and by Lemma 5.2 we may push π through it to get, for some π',

$$\vdash (P \otimes \mathsf{id}_Z)C = (\mathit{merge}_{h+k} \otimes \mathsf{id}_{Y \uplus Z})(\mathsf{id}_{\langle h,Z\rangle} \otimes \bigotimes_{i\in k} M_i)(Q_{\pi(0)} \otimes \cdots \otimes Q_{\pi(m-1)})\pi'\ .$$

Now the sequence of Q_j can be factored into products $C', C_0 \ldots, C_{k-1}$ with outer names $Z', Z_0, \ldots Z_{k-1}$, where $Z' \uplus \biguplus_i Z_i = Z$, yielding

$$\vdash (P \otimes \mathsf{id}_Z)C = (merge_{h+k} \otimes \mathsf{id}_{Y \uplus Z})\big(C' \otimes \bigotimes_{i \in k}(M_i \otimes \mathsf{id}_{Z_i})C_i\big)\pi'$$

Since each M_i has at most n ions we can apply (1) to each $(M_i \otimes \mathsf{id}_{Z_i})C_i$. Furthermore, using Lemma 5.1(2), the merge in each prime factor of C' may be combined with the outer merge. Together, these manipulations provably yield a DPNF P', as required. This concludes the inductive proof of (1) and (2).

To prove (3), note that D takes the form $(P_0 \otimes \cdots \otimes P_{n-1})\pi \otimes \alpha$. We first push π through C by Lemma 5.2; then we apply (2) to n forms $(P_i \otimes \mathsf{id}_{Z_i})C_i$; the result is then provably equal to a DDNF D' using Lemma 5.10. ■

We are now ready for our main result on provable normal forms:

Proposition 5.12 (provable normal forms) *Let E be a linear expression.*

1. *If E denotes a discrete molecule then $\vdash E = M$ for some* MDNF *M.*
2. *If E denotes a discrete prime then $\vdash E = P$ for some* PDNF *P.*
3. *If E denotes a discrete bigraph then $\vdash E = D$ for some* DDNF *D.*
4. *If G is any expression then $\vdash G = B$ for some* BDNF *B.*

Proof We first prove (3) by structural induction. The base case (elementary linear expressions) and the case for tensor product of two linear expressions are straightforward. For the case of a composition, suppose the result holds for linear E_0 and E_1; we wish to prove it for E_0E_1. By assumption we have $\vdash E_i = D_i$ for DDNF D_i $(i = 0, 1)$. Now D_1 has the form $C\pi \otimes \beta$, where C is a product of PDNFs, and we have $\vdash D_0 = D'_0 \otimes \alpha$ where D'_0 is DDNF and α is composable with β. Hence

$$\vdash E_0E_1 = D'_0C\pi \otimes \alpha\beta = D \otimes \alpha\beta$$

where D is a DDNF by Lemma 5.11, so we are done.

For (2), we first note from (1) that $\vdash E = D$, a DDNF. But since D has outer width 1 and no inner names, by inspection of the DDNF structure of we find (with the help of Lemma 5.10) that $\vdash D = P$, a PDNF, and we are done.

For (3), we first note from (2) that $\vdash E = P$, a PDNF. But since P denotes a molecule, as in the previous case we find that $\vdash D = M$, a MDNF, and we are done.

Finally, (4) follows directly from (3) and Proposition 5.8. ■

Completeness

We are now ready to prove completeness. We first need an inductive argument to prove it for linear expressions, and full completeness then follows directly.

Proposition 5.13 (linear completeness) *If E and E' are linear expressions and $E = E'$ then $\vdash E = E'$.*

Proof We first prove completeness for prime linear expressions, by induction on the number of ions in E (and hence also in E'). Assume that this holds for $< n$ ions.

First we prove the result when E and E', with n ions, denote a discrete molecule. In this case, by Proposition 5.12(1) and Proposition 4.2(1), we have provable MDNFs

$$\begin{array}{rcl} \vdash E & = & (\mathsf{K}_{\vec{x}} \otimes \mathsf{id}_Y)P \\ \vdash E' & = & (\mathsf{K}_{\vec{x}} \otimes \mathsf{id}_Y)P' \end{array}$$

where $P = P'$ are PDNFs with $< n$ ions. By the induction hypothesis we have $\vdash P = P'$, and it follows that $\vdash E = E'$.

Now we extend the result to when E and E', with n ions, denote any discrete prime. By Proposition 5.12(2) and Proposition 4.2(2) we have provable PDNFs

$$\begin{array}{rcl} \vdash E & = & (\mathit{merge}_{n+k} \otimes \mathsf{id}_Y)(\mathsf{id}_n \otimes M_0 \otimes \cdots \otimes M_{k-1})\pi \\ \vdash E' & = & (\mathit{merge}_{n+k} \otimes \mathsf{id}_Y)(\mathsf{id}_n \otimes M'_0 \otimes \cdots \otimes M'_{k-1})\pi' \end{array}$$

where M_i, M'_i, π and π' satisfy the conditions set out in Proposition 4.2(2). But these discrete molecules have no more than n ions, so we apply the above result to obtain provable equality of certain MDNFs and then, with the help of Lemma 5.1(1), provable equality for the two displayed PDNFs; hence again $\vdash E = E'$. This completes the inductive proof of the proposition for prime linear expressions.

Finally suppose that E and E' are any linear expressions. By Proposition 5.12(3) and Proposition 4.2(3) we have provable DDNFs

$$\begin{array}{rcl} \vdash E & = & (P_0 \otimes \cdots \otimes P_{n-1})\pi \otimes \alpha \\ \vdash E' & = & (P'_0 \otimes \cdots \otimes P'_{n-1})\pi' \otimes \alpha' \end{array}$$

where the P_i and P'_i are PDNFs such that $P'_i = P_i\rho_i$ and $(\rho_0 \otimes \cdots \otimes \rho_{n-1})\pi' = \pi$ for certain permutations ρ_i. But with the help of Lemma 5.10 our inductive argument shows that the former equation is provable; the latter is also provable by Lemma 5.3. It follows immediately that $\vdash E = E'$. ■

Theorem 5.14 (completeness) *If $G = G'$ then $\vdash G = G'$.*

Proof By Proposition 5.12(4) and Proposition 4.2(4) we have provable DDNFs

$$\begin{array}{rcl} \vdash G & = & (\mathsf{id}_n \otimes \omega)E \\ \vdash G' & = & (\mathsf{id}_n \otimes \omega')E' \end{array}$$

where E and E' are DDNFs such that $E = (\mathsf{id}_n \otimes \alpha)E'$ and $\omega\alpha = \omega'$ for some renaming α. But Proposition 5.13, with Lemma 5.10, shows that the former equation is provable, and the latter is also provable by Lemma 5.4. It follows that $\vdash G = G'$. ■

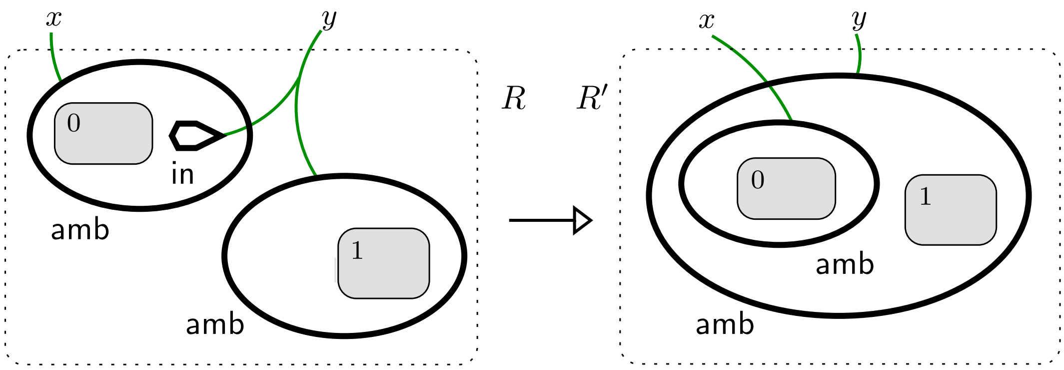


Figure 8: A reaction rule for mobile ambients

6 Programming and connected normal form

The completeness of the axiom system in Table 1 depends primarily on two things: first, that all linking can be exposed at the outermost level of an expression; second, that we have a strict symmetric monoidal category of bigraphs, with a tensor that is partial on objects. Crucial to the tensor is that it is bifunctorial, i.e. $(A_1 \otimes B_1)(A_0 \otimes B_0) = (A_1 A_0) \otimes (B_1 B_0)$; this axiom underlies most of our manipulations.

Thus the discrete normal form, DNF, has been crucial for the proof of completeness. Despite this mathematical significance, DNF is not so convenient for programming, or more generally for the practical description of systems. As a striking example of this, let us consider the calculus of mobile ambients [1], which can be formulated within bigraphs. Ambients are regions within which local activity may occur, and their mobility is capture by certain reaction rules. one of them (slightly simplified) is illustrated in Figure 8.

It means that if any part of a bigraph matches the *redex* R (with anything in the holes), then this part of the graph can be replaced by the *reactum* R' (keeping the same things in the holes). The in agent in R asks that its host ambient, the left-hand oval, should migrate into the ambient to which the in node is pointing (i.e. the right-hand oval), and should carry along all its contents with it. R' represents the result of this migration.

In this paper we are not concerned with the dynamics, only with the statics of bigraphs. So here let us use the redex R just as an example of a realistic bigraph. Let us write down what R would look like in DNF, slightly simplified:

$$R = (merge \otimes \mathsf{id}_x \otimes {}^{y}/_{\{yz\}})\big((\mathsf{amb}_x \otimes \mathsf{id}_y) merge(\mathsf{id}_1 \otimes \mathsf{in}_y 1) \otimes \mathsf{amb}_z\big) .$$

Compare this with the following alterative:

$$R = \mathsf{amb}_x(\mathsf{id}_1 \mid \mathsf{in}_y 1) \mid \mathsf{amb}_y .$$

This alternative, with a little sugaring, corresponds exactly to the standard notation of the ambient calculus. Instead of DNF it uses the *connected normal form* (CNF) which we shall shortly define in terms of the *prime parallel product* '|', which combines tensor product, merging and substitution. CNF resembles parallel composition in other process calculi; for example, the (only) reaction rule of a simplified π-calculus takes the form

$$\overline{x}y.P \mid x(z)\,.Q \longrightarrow\!\!\triangleright\; P \mid \{y/z\}Q \;.$$

Here, a sender and receiver are placed side-by-side (compare tensor product) in the same region (compare $merge$) and sharing the channel name x (compare substitution).

Parallel product

We now formally define the operation of *parallel product* '$\parallel$' on bigraphs of arbitrary width. It resembles tensor product. but allows names to be shared. Thus on interfaces it is always defined:

$$\langle m, X\rangle \parallel \langle n, Y\rangle \stackrel{\text{def}}{=} \langle m{+}n, X \cup Y\rangle \;.$$

On bigraphs with disjoint inner names it is alway defined. Suppose that $X_0 \cap X_1 = \emptyset$, and let $G_i : \langle m_i, X_i\rangle \to \langle n_i, Y_i\rangle$ for $i = 0, 1$; then we define the parallel product of G_0 and G_1 in terms of tensor product. First we disjoin their outer names by applying renamings $\alpha_i : Y_i \to Y_i'$ with $Y_0' \cap Y_1' = \emptyset$. Then setting $\sigma = \alpha_0^{-1} \cup \alpha_1^{-1}$ we define

$$G_0 \parallel G_1 \stackrel{\text{def}}{=} \sigma(\alpha_0 G_0 \otimes \alpha_1 G_1) \quad : \langle m_0{+}m_1, X_0 \uplus X_1\rangle \to \langle n_0{+}n_1, Y_0 \cup Y_1\rangle \;.$$

Just like $\otimes$ this product is associative, with unit id_ϵ. It is defined more often than $\otimes$, and when both products are defined they are equal. But unlike $\otimes$ it is not bifunctorial; instead it satisfies

$$(G_1 \parallel H_1)(G_0 \otimes H_0) \;=\; (G_1 G_0) \parallel (H_1 H_0)$$

which suggests that, algebraically, $\parallel$ does not stand well by itself; it needs $\otimes$ to be present as an auxiliary.

Before defining CNF we need another operator, the *prime parallel product* '$\mid$', which always creates a prime, even on non-prime arguments. (In contrast, if G_0 and G_1 are prime then $G_0 \parallel G_1$ has width 2). It is defined simply by

$$G_0 \mid G_1 \;\stackrel{\text{def}}{=}\; merge(G_0 \parallel G_1) \;.$$

It is again associative; it has the prime 1 as a unit. Again, it is not bifunctorial but satisfies

$$(G_1 \mid H_1)(G_0 \otimes H_0) \;=\; (G_1 G_0) \mid (H_1 H_0) \;.$$

This operator agrees strongly with the homonymous operator in process calculi; indeed, when we translate either the π-calculus or the calculus of mobile ambients into bigraphs, '$\mid$' in the calculus is encoded by '$\mid$' in bigraphs. The difference is that those calculi do not in general have categorical composition; their processes correspond to those bigraphs whose inner face is ϵ. However, the dynamic theory of bigraphs [9] allows us to *derive* labelled transition systems where, in each transition $G \xrightarrow{L}\!\triangleright G'$, the label L is itself a bigraphs (normally a little one) such that $L \circ G$ is defined. Treating labels as composable entities is essential to the uniform behavioural theory that bigraphs provide.

Connected normal form

The key distinction of the parallel products is that they allow sharing of names. This sharing is induced by the substitution σ used in their definition. Thus, by using them instead of tensor product, we are pushing substitutions *inwards* as far as possible. The key to connected normal form, then, is to push *all* linking inwards as far as possible, including closures. This is a close parallel to the good advice often given to programmers, to declare their variables in as small a scope as possible. With this in mind, we are ready for our last definition:

Definition 6.1 (connected normal forms) There are three kinds of CNF: MCNFs M for molecules, PCNFs P for primes and BCNFs B for bigraphs:

$$\begin{array}{rl} \text{MCNF:} & M ::= (/Z \mid \mathsf{id}_1)(\mathsf{K}_{\vec{x}} \mid \mathsf{id}_Y)P \\ \text{PCNF:} & P ::= (/Z \mid \mathsf{id}_1)(\mathsf{id}_n \mid M_0 \mid \cdots \mid M_{k-1})\pi \\ \text{BCNF:} & B ::= (/Z \parallel \mathsf{id}_n)(\sigma \parallel (P_0 \parallel \cdots \parallel P_{n-1})\pi)\ . \end{array}$$

The names $\vec{x}$ need not be distinct in the MCNF. Moreover, in each case any closed name $z \in Z$ must occur in at least two members of the ensuing product ($\parallel$ or $\mid$). ■

Many of the technical properties of DNF are shared by CNF. First, every molecule (resp. prime, bigraph) can be expressed by a MCNF (resp. PCNF, BCNF). Second, these expressions are unique up to certain isomorphisms, which are almost exactly as described in Proposition 4.2.

However, we leave open the question of a complete axiomatisation expressed in terms of the parallel products instead of the tensor product. Even if it exists it is unlikely to be as simple, because the bifunctorial property of the tensor product is absent. This does not alter the fact that the CNFs appear to be simpler. I conjecture that a programming language for bigraphs may well be based firmly on CNFs but with added notational convenience, just as existing process calculi are based. Moreover, the conversion to CNF is not hard and can be done whenever necessary for theoretical analysis.

7 Related and further work

The algebraic formulation of bigraphs arises from a category in which the objects are interfaces and the arrows are graphs, following Lawvere's paradigm for algebraic theories [11]. It is interesting and non-trivial to explore the relationship between this treatment —particularly as it affects dynamic theory— with that of graph-rewriting using the double pushout construction [3], where the objects are graphs and the arrows are graphs embeddings. The Lawvere approach is closer to the algebraic tradition in process calculi; we are keen to adopt it both for comparison with these calculi and because the algebraic approach to concurrent processes has been fruitful. But a link between the two approaches has been identified [4], and indeed the techniques for deriving behavioural congruences has been transferred [5] from the algebraic framework to the embedding framework. More remains to be done to discover the relative benefits of the approaches, both in allowing different kinds of graph and in analysing real systems in practice.

Process calculi, extended with stochastics, are becoming successful in modelling biological processes, for example signal transduction in cells [20]. Recently, as one might expect, explicitly topographic models such as mobile ambients are being employed in this way [19]. This link with biology is encouraging, and also valuable as a test for the wider applicability of process models arising from computer science. Also, as our opening example suggests, there is a strong incentive to build topographical calculi that can model the phenomena of ubiquitous computing, where mobile communicating automata (both macro- and microscopic) will abound [10].

Let us now turn to possible variations and developments in bigraph theory. The present definition is not canonical. For example, for some purposes we may wish to consider bigraphs whose regions (nodes) may overlap one another. For example, one node may represent a physical location, say San Francisco, and an overlapping node may represent the University of California which has separate campuses in many Californian cities. Can bigraphs respond to this challenge? At one level, it is easy; we simply generalise place graphs from forests of trees to directed (acyclic) graphs. However, the impact of this change on our algebra is not obvious. It should be remarked, though, that the computational models we have studied do not need this extra feature.

Another variation is to consider *polygraphs*, where there may be more than two orthogonal structures. Indeed this may be the way to cope with overlapping nodes; for we may consider nodes to be structured both by nesting of *physical* regions (such as San Francisco) and by nesting of *virtual* regions (such as the University of California), without either form of locality constraining the other. Indeed the dynamic theory of bigraphs [10] does not exclude this, since it relies on categorical concepts developed separately for each of the orthogonal structures. But again, the impact on algebraic theory is not yet clear.

Within our present theory, much refinement is possible and is being examined. The first is to do with local names, or localised links. Once the theory is developed one can constrain the independence of linking and placing by introducing *bound* names, those that confine a link to within a certain place. This is needed for modelling established process calculi, and has turned out [8, 9] that the refined theory can readily be embedded in the pure theory. We conjecture that the present algebraic theory can be simply adapted to the refined theory. From the algebraic viewpoint an even more obvious refinement is to introduce *sorting*, allowing both the regions and the names of to be many-sorted. This has already been employed in modelling Petri nets [17] and the full π-calculus [7].

To conclude: the algebraic treatment of mobile processes with explicit regions has considerably potential, and the purpose of the present paper has been to apply traditional algebraic methodology to the static structure of such processes, and thus to provide a firm basis for the supervening dynamical theory.

References

[1] Cardelli, L. and Gordon, A.D. (2000), Mobile ambients. Foundations of System Specification and Computational Structures, LNCS 1378, pp140–155.

[2] 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://pauillac.inria.fr/~leifer`.

[3] Ehrig, H. (1979) Introduction to the theory of graph grammars. Graph Grammars and their Application to Computer Science and Biology, LNCS 73, Springer Verlag, pp1–69.

[4] Ehrig, H. (2002) Bigraphs meet double pushouts. EATCS Bulletin 78, October 2002, pp72–85.

[5] Ehrig, H. and König, B. (2004) Deriving bisimulation congruences in the DPO approach to graph rewriting. Proc. FOSSACS 2004, Barcelona.

[6] Gardner, P.A. (2000), From process calculi to process frameworks. Proc. CONCUR 2000, 11th International Conference on Concurrency Theory, pp69–88.

[7] Jensen, O.H. (2004). Forthcoming PhD Thesis.

[8] Jensen, O.H. and Milner, R. (2003), Bigraphs and transitions. In 30th SIGPLAN-SIGACT Symposium on Principles of Programming Languages.

[9] Jensen, O.H. and Milner, R. (2004), Bigraphs and mobile processes (revised). Technical Report UCAM-CL-TR-580, University of Cambridge Computer Laboratory.

[10] Kwiatkowska, M., Milner, R. and Sassone, V. (2004), Theories for the global ubiquitous computer. Bulletin of thew European Association for Theortical Computer Science, to appear.

[11] Lawvere, F.W. (1963), Functorial semantics of algebraic theories. Proc. Nat. Acad. Sci. 50, pp869–872.

[12] Leifer, J.J. (2001), Operational congruences for reactive systems. PhD Dissertation, University of Cambridge Computer Laboratory. Distributed in revised form as Technical Report 521. Available from `http://pauillac.inria.fr/~leifer`.

[13] Leifer, J.J. and Milner, R. (2000), Deriving bisimulation congruences for reactive systems. Proc. CONCUR 2000, 11th International Conference on Concurrency theory, pp243–258. Available at `http://pauillac.inria.fr/~leifer`.

[14] Milner, R. (1996), Calculi for interaction. Acta Informatica 33, pp707–737.

[15] Milner, R. Bigraphical reactive systems: basic theory. Technical Report 503, University of Cambridge Computer Laboratory (2001). Available from `http://www.cl.cam.ac.uk/users/rm135`.

[16] Milner, R. (2001) Bigraphical reactive systems. Proc. 12th International Conference on Concurrency Theory, LNCS2154, pp16–35.

[17] Milner, R. (2004), Bigraphs for Petri nets. To appear in Proc. Advance Course in Petri Nets, Eichstätt, 2003, Lecture Notes in Computer Science, Springer Verlag.

[18] Milner, R., Parrow, J. and Walker D. (1992), A calculus of mobile processes, Parts I and II. Journal of Information and Computation, Vol 100, pp1–40 and pp41–77.

[19] Regev, A., Panina, E., Silverman, W., Cardelli, L. and Shapiro, E. (2004), BioAmbients: an abstraction for biological compartments. Theoretical Computer Science, to appear.

[20] Priami, C., Regev, A., Silverman, W. and Shapiro, E. (2001), Application of a stochastic name-passing calculus to rerpesentation and simulation of molecular processes. Information Processing Letters 80, pp25–31.

[21] Sewell, P. (1998), From rewrite rules to bisimulation congruences. Proc CONCUR'98, LNCS 1466, pp269–284. Full version to appear in Theoretical Computer Science 272.