Nordic Journal of Computing

Axiomatizing Binding Bigraphs

T.C. Damgaard
IT University of Copenhagen
Denmark
tcd@itu.dk

L. Birkedal
IT University of Copenhagen
Denmark
birkedal@itu.dk

Abstract. We axiomatize the congruence relation for binding bigraphs and prove that the generated theory is complete. In doing so, we define a normal form for binding bigraphs, and prove that it is unique up to certain isomorphisms.

Our work builds on Milner's axioms for pure bigraphs. We have extended the set of axioms with five new axioms concerned with binding, and we have altered some of Milner's axioms for ions, because ions in binding bigraphs have names on both their inner and outer faces. The resulting theory is a conservative extension of Milner's for pure bigraphs.

ACM CCS Categories and Subject Descriptors: D.3.1. Formal Definitions and Theory, F.3.2. Process Models.

Key words: graphical models of computation, bigraphs, axioms for static congruence.

1. Introduction

Over the last decade, Robin Milner and co-workers have developed a theory of bigraphical reactive systems, see [9, 12, 13]. Bigraphical reactive systems (BRSs) provide a graphical model of computation in which both locality and connectivity are prominent. In essence, a *bigraph* consists of a *place graph*, a forest, whose nodes represent a variety of computational objects; and a *link graph*, which is a hyper graph connecting ports of the nodes. Bigraphs can be reconfigured by means of *reaction rules*. A *bigraphical reactive system* consists of set of bigraphs and a set of reaction rules.BRSs have been developed with two principal aims: (1) to model ubiquitous systems by focusing on mobile connectivity (the link graph) and mobile locality (the place graph), and (2) to provide a unification of existing theories by developing a general theory, in which many existing calculi for concurrency and mobility may be represented, with a uniform behavioural theory. The latter is achieved by representing the dynamics of bigraphs by reaction rules from which a labelled transition system may be derived in such a way that the associated bisimulation relation is a congruence. The unification has recovered existing behavioural theories for the π-calculus [9], the ambient calculus [10], and has contributed to that for Petri nets [11]. Thus the evaluation of the second aim has so far been encouraging. In [3] Birkedal et al. initiate an evaluation of the first aim, in particular it is shown how to give bigraphical models of context-aware systems.

As suggested and argued in [9, 2, 1, 3] it would be very useful to have an implementation of the dynamics of bigraphical reactive systems to allow experimentation and simulation. In the Bigraphical Programming Languages research project at the IT University, we are working towards such an implementation.

An implementation of bigraphical reactive systems must, of course, work on some data structure representing bigraphs. An obvious possibility is to represent bigraphs by *bigraphical expressions* that denote bigraphs. This is particularly feasible if (1) the bigraphical expressions are defined inductively (by a grammar, say), such that algorithms may operate inductively on the representation; and (2) there are normal forms for bigraphical expressions and axioms for determining whether two bigraphical expressions denote the same bigraph, such that algorithms may operate on normal form representations, and may be founded on principles of equational reasoning. There *is* such an axiomatization of the so-called *pure* bigraphs with these properties [12]. In the present paper we extend the axiomatization for pure bigraphs to *binding bigraphs*, a wider class of bigraphs better suited for the representation of calculi and systems involving binding, e.g., the π-calculus, and prove that our axiomatization has the above mentioned desired properties. In particular, we prove the axiomatization complete and prove that our notion of normal form is unique up to certain specified isomorphisms. Our axiomatization is a conservative extension of Milner's.

For reasons of brevity, we refer the reader to the papers cited above for more background information and motivation than can be included here. In particular, we shall need to assume some familiarity with pure and binding bigraphs as described in [9] and with the axiomatization of pure bigraphs [12] — we do, however, include an informal description of bigraphs in the following section and we have included the formal definition of binding bigraphs in Appendix A.

The remainder of the paper is organized as follows. In the following section we introduce bigraphs by example. In Section 3 we define elementary forms of bigraphs and arrive at a semantic normal form theorem, which expresses how every bigraph may be decomposed into a composite of elementary forms. In Section 4 we present our term language for binding bigraphs and the accompanying equational theory. We arrive at a theorem which states soundness and compleness of the equational theory. We present some examples of bigraphs and their corresponding normal forms in Section 5 — we recommend that the reader refers to these examples from time to time when reading the earlier more technical sections. We comment on some related and further work in Section 6. Finally, Appendix A contains a summary of the definitions of binding bigraphs. We have omitted detailed proofs from this paper, they can be found in the companion technical report [5].

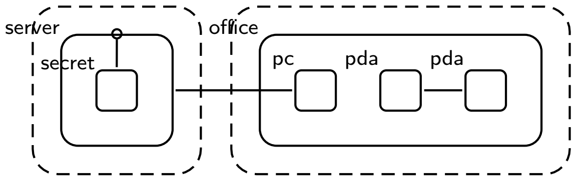

Figure 2.1: A – a bigraph model of an office in a building

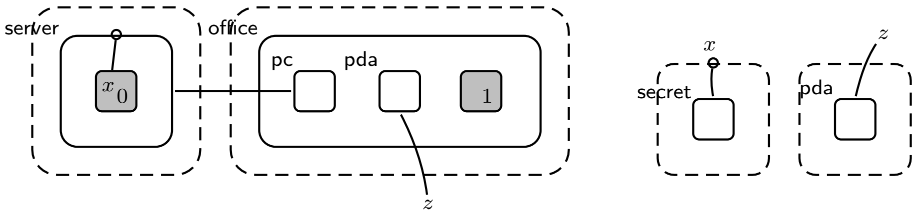

Figure 2.2: B and C – bigraphs that compose to form A

2. Bigraphs by Example

We introduce the most basic terminology and properties for bigraphs, by giving a small example of a bigraph. We refer the reader to Appendix A for all formal definitions.

The bigraph A is bigraph model of an office containing a pc and two pdas. The pc is linked (supposedly by some kind of network connection) to server containing a secret located somewhere else. We say that A consists of **roots** (dashed boxes), **nodes** (solid boxes), and **links** (lines). Each node has a **control** written beside it. The control indicates the number and type of ports for linkage on the node. Ports can be either **free** or **binding** — the latter indicated by circular attachments.

Bigraphs can contain **sites** (sometimes called holes), and/or inner or outer names. The bigraph B has two sites, numbered 0 and 1, and two inner names, x **located** at site 0 and z **global** (i.e., not located). C has two outer names, x located at its first root, and x global.

We can compose B and C by plugging the sites of B with the roots of C. The bigraphs B and C **compose** to form A. We write $A = B \circ C$. Bigraph A is said to be **ground** as it has no holes or inner names.

Binding bigraphs enforce a **scope** discipline on linkage connected to a binding port: All **peers** (names or ports) linked to a binding port or located outer name, must be nested within the node or root (see Definition 13).

Not all bigraphs are composable. B and C composes exactly, because C has a root, outer name (local and global), for each corresponding site and inner name (local and global) of B. The **interfaces** of a bigraph registers this, and hence determines which bigraphs can be composed. We write $B : \langle 2, (\{x\}, \emptyset), \{x, z\}\rangle \rightarrow \langle 2, (\emptyset, \emptyset), \emptyset\rangle$ and $C : \langle 0, (\emptyset), \emptyset\rangle \rightarrow \langle 2, (\{x\}, \emptyset), \{x, z\}\rangle$.

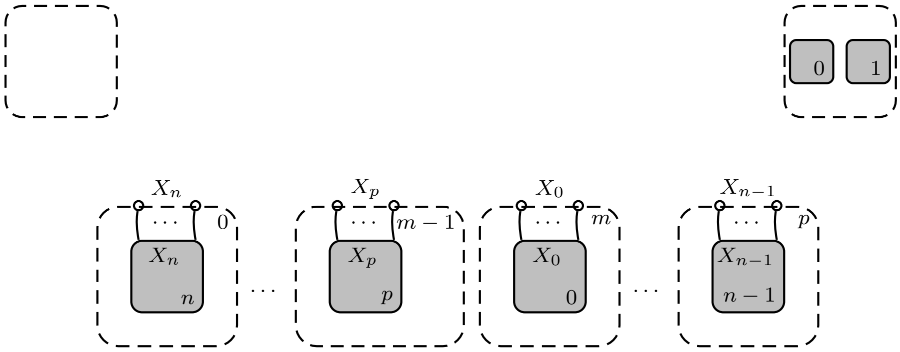

Figure 3.3: 1, *join*, and $\gamma_{m,n,(\vec{X},\vec{Z})}$ (using the abbreviation $p = m + n - 1$)

We can also combine bigraphs by an associative **tensor product** (denoted by $\otimes$), which works simply by juxtaposition of roots. For tensor product we require only that both inner and outer names be disjoint.

Finally, in the following we will be particularly concerned with three classes of bigraphs — **prime** bigraphs are those with only a single root, and only local inner names. For **discrete** bigraphs all linkage upon global names is one-one while **name-discrete** bigraphs, are those where *all* linkage upon all names is one-one (refer to Definition 14 for the full definition of discreteness).

For more involved examples of bigraphical models including dynamics, we refer the reader to the tech report [6].

3. Elementary Bigraphs and Normal Form

We start by defining **placings** corresponding closely to the placings defined for pure bigraphs in [12]. We shall use placings to define the class of terms for bigraphs that denote place graphs paired with identities on local names.

$$\begin{array}{rcll} 1 & : & \epsilon \to 1 & \text{a barren root,} \\ \mathit{join} & : & 2 \to 1 & \text{join two sites,} \\ \gamma_{m,n,(\vec{X},\vec{Z})} & : & \multicolumn{2}{l}{\langle m+n, \vec{X}\vec{Z}, \{\vec{X}\} \uplus \{\vec{Z}\}\rangle \to \langle m+n, \vec{Z}\vec{X}, \{\vec{X}\} \uplus \{\vec{Z}\}\rangle} \\ & & & \text{transpose } m \text{ with } n \text{ places preserving names.} \end{array}$$

Note that 1 and *join* are defined exactly as for pure bigraphs, while $\gamma_{m,n,(\vec{X},\vec{Z})}$ lets a set of local inner names be linked to corresponding outer names, in the only way allowed by the scope rule (see Definition 13).

We use π and ρ to range over **permutations**, placings generated by composition and tensor product from $\gamma_{m,n,(\vec{X},\vec{Z})}$.

For $I_i = \langle m_i, \vec{X^i_B}, \{\vec{X^i_B}\} \uplus X^i_F \rangle$ $(i \in \{0,1\})$ we define

$$\gamma_{I_0,I_1} \stackrel{\text{def}}{=} \gamma_{m_0,m_1,(\vec{X^0_B},\vec{X^1_B})} \otimes \mathsf{id}_{X^0_F} \otimes \mathsf{id}_{X^1_F}.$$

Using *join* we define the bigraph $merge_m$ that joins m sites:

Definition 1 (merge)**.** *For all $m \geq 0$ we define $merge_m : m \to 1$ recursively, by*

$$\begin{aligned} merge_0 &\stackrel{\text{def}}{=} 1 \\ merge_{m+1} &\stackrel{\text{def}}{=} join(\mathsf{id}_1 \otimes merge_m). \end{aligned}$$

Note that $merge_1 = \mathsf{id}_1$ and thus $merge_2 = join$.

A **linking** is a (pure) link graph $X \to Y$ that has no nodes. All linkings can be expressed in terms of the following two kinds:

$$\begin{aligned} /x &: x \to \epsilon && \text{closure,} \\ y/X &: X \to y && \text{substitution } x \mapsto y, \text{ for all } x \in X. \end{aligned}$$

A closure closes a single link. For $X = \{x_0, \ldots, x_{k-1}\}$ and $k > 0$ we define a multiple closure $/X \stackrel{\text{def}}{=} /x_0 \otimes \cdots \otimes /x_{k-1}$. For $Y = \{y_0, \ldots, y_{k-1}\}$, $k > 0$, and disjoint sets $X_0, \ldots, X_{k-1}$ we define a multiple substition

$$\vec{y}/\vec{X} \stackrel{\text{def}}{=} y_0/X_0 \otimes \cdots \otimes y_{k-1}/X_{k-1}.$$

Note that a substitution need not be surjective (i.e., we allow $X = \emptyset$), thus the dual of closure – name introduction $y : \epsilon \to y$ – is a substitution. A **renaming** is a bijective (multiple) substitution, i.e., each X_i above is a singleton. A **wiring** is a bigraph with zero width (and hence no local names) generated by composition and tensor of $/x$ and y/X.

We let ω range over wirings, σ range over (multiple) substitutions and α and β range over renamings. Often we do not distinguish notationally between a name and the singleton set containing the name. With this convention $\vec{y}/\vec{x}$ is a renaming when $\vec{y} = y_0, \ldots, y_{k-1}$ and $\vec{x} = x_0, \ldots, x_{k-1}$, for some k.

A **simple concretion** is a discrete prime which maps a set X of local inner names severally to equally named global outer names.

$$\ulcorner X \urcorner : (X) \to \langle X \rangle \quad \text{concretion.}$$

Note that a special case of a simple concretion is $\mathsf{id}_1 = \ulcorner \emptyset \urcorner$.

An **abstraction** $(X)-$ is a *construction*, defined on every prime P that localizes a subset of the global names of P. For every prime $P : I \to \langle (Y_B), Y \rangle$, let

$$(X)P \quad : I \to \langle (Y_B \uplus X), Y \rangle \quad \text{abstraction on } P,$$

where $X \subseteq Y \setminus Y_B$.

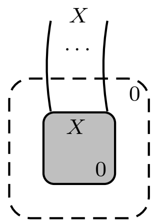

Figure 3.4: $\ulcorner X \urcorner$

Note that the scope rule is necessarily respected since the inner face of P is required to be local as P is prime. Abstractions are in some sense dual to concretions, and the axioms concerning abstraction and concretion reflect this (see axioms (B2) and (B3) in Table I)

Using abstraction we can express concretions in the sense of [9]: We define a general **concretion** $\ulcorner Y \urcorner^X : \langle 1, (X \uplus Y), X \uplus Y \rangle \to \langle 1, (X), X \uplus Y \rangle$ in terms of a simple concretion and abstraction as $\ulcorner Y \urcorner^X \stackrel{\text{def}}{=} (X)\ \ulcorner X \uplus Y \urcorner$. Towards succinct statement of the normal form, we define $\ulcorner \alpha \urcorner \stackrel{\text{def}}{=} (\alpha \otimes \mathsf{id}_1)\ulcorner X \urcorner$ (where $\alpha : X \to$).

With the help of linkings we get **local wirings** — bigraphs that by composition can change the linkage of local names. We define a **local renaming** (for vectors of names $\vec{y}$ and $\vec{x}$, s.t. $|\vec{y}| = |\vec{x}|$) by $(\vec{y})/(\vec{x}) \stackrel{\text{def}}{=} (\vec{y})((\vec{y}/\vec{x} \otimes \mathsf{id}_1)\ulcorner \{\vec{x}\} \urcorner)$. We extend this notation to multiple substitutions and define $(\vec{y})/(\vec{X}) \stackrel{\text{def}}{=} (\vec{y})((\vec{y}/\vec{X} \otimes \mathsf{id}_1)\ulcorner X \urcorner)$ (for $X = \{\vec{X}\}$).

Just as plain substitutions can introduce idle global names, local substitutions can introduce idle local names when their underlying global substitution is not surjective (e.g., $(y)/(\emptyset)$).

We let $\alpha^{\mathbf{loc}}$ and $\sigma^{\mathbf{loc}}$ range over local renamings and substitutions, respectively. We shall need to take the preimage of a local substitution $\sigma^{\mathbf{loc}}$ of a vector of namesets $\vec{X}$. Formally:

Definition 2 (Preimage of a local wiring)**.** *Let $\sigma^{\mathbf{loc}}_{\mathbf{u}}$ be the link map (which is a function) of $\sigma^{\mathbf{loc}}$. For a set of names X, define $(\sigma^{\mathbf{loc}})^{-1}(X)$ to be the preimage $(\sigma^{\mathbf{loc}}_{\mathbf{u}})^{-1}(X)$ and define $(\sigma^{\mathbf{loc}})^{-1}(\vec{X})$ to be the vector of namesets resulting from taking the preimage of $\sigma^{\mathbf{loc}}$ pointwise for each set in $\vec{X}$.*

We can generate all isomorphisms in the category of binding bigraphs using permutations π, renamings α, and local renamings $\alpha^{\mathbf{loc}}$(see [9, Proposition 9.2b] for the definition of isomorphism in the category of binding bigraphs):

Proposition 1. *Every binding bigraph isomorphism, $\iota : \langle m, \vec{Z}, \{\vec{Z}\} \uplus U \rangle \to \langle m, \vec{X}, \{\vec{X}\} \uplus Y \rangle$ (of width m) may be expressed in the following form*

$$\iota = (\pi \otimes \alpha)(\nu_0 \otimes \cdots \otimes \nu_{m-1} \otimes \mathsf{id}_U)$$

where these requirements hold:

- $m = |\vec{X}| = |\vec{Z}|$,

- ∘ $\alpha : U \to Y$,
- ∘ $\forall i \in m : \nu_i = (\vec{x_i})/(\vec{z_i})$ *for* $\vec{X} = (\{\vec{x}_0\}, \ldots, \{\vec{x}_{m-1}\})$, *and* $\vec{Z} = (\{\vec{z}_0\}, \ldots, \{\vec{z}_{m-1}\})$.

For a control $K : b \to f \in \mathcal{K}$, let $\vec{y}$ be a sequence of distinct names, and $\vec{X}$ a sequence of sets of distinct names, s.t. $|\vec{X}| = b$ and $|\vec{y}| = f$.

A **binding ion** $K_{\vec{y}(\vec{X})} : \langle 1, (X), X\rangle \to \langle 1, (\emptyset), Y\rangle$ is a prime bigraph with a single node of control K with free ports linked severally to global outer names $\vec{y}$, and each binding port $i \in b$ linked to all local inner names in X_i. Figure 3.5 shows a binding ion.

$$K_{\vec{y}(\vec{X})} \quad : \quad (X) \to \langle Y\rangle \quad \text{a binding ion}$$

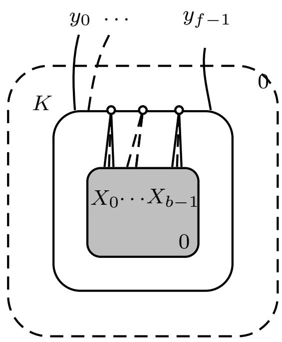

Figure 3.5: A binding ion

This definition of binding ion is a straightforward generalization of the **free discrete ion** defined in [9, Chapter 11]. We can recapture the latter by requiring every set in X to be a singleton. When $\vec{X} = (\{x_0\}, \ldots, \{x_{b-1}\})$, we overload our notation and write $K_{\vec{y}(\vec{x})}$ to mean a free discrete ion.

Definition 3. *For any name-discrete prime* $P : I \to \langle 1, (X), X \uplus Z\rangle$ *and ion* $K_{\vec{y}(\vec{X})}$*, we define a* ***free discrete molecule*** *as*

$$(K_{\vec{y}(\vec{X})} \otimes \mathsf{id}_Z)P : I \to \langle 1, (\emptyset), \{\vec{y}\} \uplus Z\rangle$$

Note that even though we use the more general binding ion in the definition above, our definition of free discrete molecule is equal to the one given in [9, Chapter 11], in the sense that it covers the same set of bigraphs.

As P in the above definition is discrete and prime it is easily seen that M is also discrete and prime. In fact:

Proposition 2. *A free discrete molecule is a name-discrete prime bigraph with a single outermost node.*

This proposition relies on both name-discreteness and discreteness being preserved by composition and tensor (Lemma 13). Vice versa, we have:

Proposition 3. *Any free discrete prime bigraph with a single outermost node is a free discrete molecule.*

3.1 A Normal Form for Binding Bigraphs

In the following section we present our binding discrete normal form theorem for graphs. This *semantic* theorem states that every binding bigraph can be decomposed in certain ways. We shall use it as the basis for the establishment of a corresponding *syntactic* definition of normal form for our term language for binding bigraphs, which we introduce in Section 4.

We aim to base our normal form on a variant of discreteness, as in [12], simply as this allows a clean separation between the constituent components of a bigraph. Our main aim is to prove completeness for an equational theory over a term language for binding bigraphs. To that end it will be central to formulate an inductive property of expressions that characterizes our chosen variant of discreteness syntactically. Alas, discreteness is not preserved under composition with abstractions and concretions. Indeed, consider a discrete bigraph D with width n. $(\bigotimes_{i<n} \ulcorner X_i \urcorner)D$ is not discrete, if D is not name-discrete. Conversely, given a nondiscrete prime $P : I \to \langle (X), X \uplus Y \rangle$, $(Y)P : I \to (X \uplus Y)$ is discrete. Hence, we turn to name-discreteness.

Recall that a bigraph is name-discrete (Definition 14) if every free link is an outer name and has exactly one point, and every bound link is either an edge, or (if it is an outer name) has exactly one point. This is a simple specialization of the discreteness property. As a consequence, it is easy to verify that both abstraction and composition with concretions preserve both name-discreteness and non-name-discreteness. Name-discreteness still allows arbitrary linking upon *bound* edges, and exactly for that reason, we have chosen to take the binding ion (as defined above) as a constant in our term language. Syntactically, this allows us to restrict the usage of substitutions to define a simple inductive property that characterizes name-discreteness.

Theorem 1 (Semantic binding discrete normal form)**.**

(1) Any free discrete molecule $M : I \to \langle 1, \{\vec{y}\} \uplus Z \rangle$ can be expressed as

$$\left(K_{\vec{y}(\vec{X})} \otimes \mathsf{id}_Z\right) P$$

where $P : I \to \langle 1, (\{\vec{X}\}), \{\vec{X}\} \uplus Z \rangle$ is a name-discrete prime.
This expression is unique up to renaming of the local names on the innerface of the ion, and (correspondingly) on the outer face of prime P. Hence, any other such expression for M takes the form

$$\left(K_{\vec{y}(\vec{X}')} \otimes \mathsf{id}_Z\right) P'$$

where the following requirements hold:

- *there exists a local renaming* $\alpha^{\mathbf{loc}} : (\{\vec{X}'\}) \to (\{\vec{X}\})$ *s.t.* $K_{\vec{y}(\vec{X})}\alpha^{\mathbf{loc}} = K_{\vec{y}(\vec{X}')}$, *and*
- $P = (\alpha^{\mathbf{loc}} \otimes \mathsf{id}_Z)P'$.

(2) *Any name-discrete prime* $P : I \to \langle 1, (Y_B), Y\rangle$ *may be expressed as*

$$(Y_B)\left(merge_{n+k} \otimes \mathsf{id}_Y\right)(\ulcorner\alpha_0\urcorner \otimes \cdots \otimes \ulcorner\alpha_{n-1}\urcorner \otimes M_0 \otimes \cdots \otimes M_{k-1})\,\pi$$

where every $M_i : J_i \to \langle Y_i^{\mathbf{M}}\rangle$ *is a free discrete molecule, and for renamings* $\alpha_i : X_i \to Y_i^{\mathbf{C}}$, *we have* $Y = (\biguplus_{i\in n} Y_i^{\mathbf{C}}) \uplus \biguplus Y_i^{\mathbf{M}}$.
The expression for P *is unique up to reordering of the concretions and molecules, and the ordering of the sites inside the molecules; the permutation changes accordingly to preserve the innerface. Formally, any other such expression for* P *takes the form*

$$(Y_B)\left(merge_{n+k} \otimes \mathsf{id}_Y\right)\left(\ulcorner\alpha'_0\urcorner \otimes \cdots \otimes \ulcorner\alpha'_{n-1}\urcorner \otimes M'_0 \otimes \cdots \otimes M'_{k-1}\right)\pi'$$

where the following requirements hold:

- *There exist permutations* ρ, ρ_i $(i \in k)$, ρ', *s.t.*
  - $\ulcorner\alpha'_i\urcorner = \ulcorner\alpha_{\rho(i)}\urcorner$
  - $M'_i = M_{\rho(i)}\rho_i$,
  - $(\mathsf{id}_{(X'_0)} \otimes \cdots \otimes \mathsf{id}_{(X'_{n-1})} \otimes \rho_0 \otimes \cdots \otimes \rho_{k-1})\pi' = \rho'\pi$.
- *Furthermore, let* $\vec{l}$ *denote the vector of inner widths of the product* $((\alpha_0 \otimes \mathsf{id}_1)\ulcorner X_0\urcorner \otimes \ldots \otimes (\alpha_{n-1} \otimes \mathsf{id}_1)\ulcorner X_{n-1}\urcorner \otimes M_0 \otimes \cdots \otimes M_{k-1})$, *let* $\vec{X}' = (X'_0, \ldots, X'_{k-1})$, *and let* $\vec{X} = (X_0, \ldots, X_{n-1})$.
  Then ρ' *is determined uniquely by* ρ, $\vec{l}$, $\vec{X}$, *and* $\vec{X}'$ *as* $\rho' = \overline{\rho}_{\vec{l},\vec{X}'\vec{X}}$ *as defined in Lemma 2.*

(3) *Any name-discrete bigraph* D *(of outer width* n*) can be expressed as*

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

where every P_i *is a name-discrete prime,* α *is a renaming, and* π *is a permutation.*
This expression is unique up to reordering of the sites in the primes; the permutation changes accordingly to preserve the innerface. Hence, any other such expression of D *takes the form*

$$\left(P'_0 \otimes \cdots \otimes P'_{n-1}\right)\pi' \otimes \alpha$$

where there exists permutations ρ_i, $(i \in n)$, *s.t.* $P'_i = P_i\rho_i$, *and* $(\rho_0 \otimes \cdots \otimes \rho_{n-1})\pi' = \pi$.

(4) Any bigraph $G : I \to \langle n, \vec{Y_B}, \{\vec{Y_B}\} \uplus Y_F \rangle$ *can be expressed as*

$$\left(\bigotimes_{i<n} (\vec{y_i})/(\vec{X_i}) \otimes \omega \right) D$$

where $D : I \to \langle n, \vec{X}, X \uplus Z \rangle$ *is name-discrete,* $\omega : Z \to Y_F$ *is a wiring, and* $\bigotimes_{i<n} (\vec{y_i})/(\vec{X_i}) : (\vec{X}) \to (\vec{Y_B})$ *is a local substitution of width* n *on the bound names of* D.

The expression is unique up to (local and global) renamings on the innerface of the wiring and (correspondingly) on the outerface of D. *Hence, any other such expression of* G *takes the form*

$$\left(\bigotimes_{i<n} (\vec{y_i})/(\vec{X'}_i) \otimes \omega' \right) D'$$

where there exists a renaming α *s.t.* $\omega' = \omega\alpha$, *and* n *local renamings* $\alpha_i^{\mathbf{loc}} : (\vec{X'}_i) \to (\vec{X_i})$, *s.t.* $(\bigotimes_{i<n} (\vec{y_i})/(\vec{X_i})) \bigotimes_{i<n} \alpha_i^{\mathbf{loc}} = (\bigotimes_{i<n} (\vec{y_i})/(\vec{X'}_i))$, *and* $\left(\bigotimes_{i<n} \alpha_i^{\mathbf{loc}} \otimes \alpha \right) D' = D$.

Furthermore, for every class of expressions the expression given is well defined and generates only *bigraphs of the appropriate type.*

See [5] for a proof of the theorem. The proof is simply a detailed analysis of the structure of possible decompositions of binding bigraphs.

4. Binding Bigraph Expressions and Axioms

The set of **binding bigraph expressions** is defined as the smallest set of expressions built by composition, tensor product, and abstraction (on primes) from identities and the constants we have just introduced:

$$1 \qquad join \qquad \gamma_{m_0,m_1,(\vec{X_0},\vec{X_1})} \qquad /x \qquad y/X \qquad \ulcorner X \urcorner \qquad K_{\vec{y}(\vec{X})}$$

Each expression (implicitly) has two interfaces of the form $\langle m, \vec{X}, Y \rangle$ which determine when tensor product, composition, and abstraction are well defined (according to the requirements stated formally in Appendix A). The interface and the bigraph an expression denotes can be determined by induction. As usual, we write $\vDash E = F$ to mean that the expression $E = F$ is **valid**; and $\vdash E = F$ if the equation is **provable**.

In [12] Milner stated and proved a set of axioms complete for pure bigraph expressions. We extend that result and prove the set of axioms in Table I complete for binding bigraph expressions. Every pure bigraph expression as defined by Milner [12] trivially corresponds to a binding bigraph expression as defined above. Our axiomatic theory is a conservative extension of Milner's in the sense that any two pure bigraph expressions are provably equal in Milner's theory iff the corresponding expressions are provably equal in our theory. (Formally, this is easy to prove using soundness and completeness of the two theories and the fact that the embeddings of pure bigraphs into binding bigraphs and pure bigraph expressions into binding bigraph expressions are both full and faithful). We proceed by defining and proving the theory complete for increasingly larger classes of expressions.

Note that as tensor product is defined only when name sets of the interfaces are disjoint, and as abstraction is defined only on prime bigraphs with the abstracted names in the outer face, we only require the equations to hold when both sides are defined.

4.1 Preliminaries

Lemma 1 (Wiring commutes with all binding bigraph expressions)**.** *For all bigraph expressions G and for all wirings $\omega \vdash G \otimes \omega = \omega \otimes G$.*

By essentially iterating axiom C9, we can push a permutation "through" a product of primes, permuting the order in which they appear in the product, and producing a permutation that reorders the sites in the primes to preserve the inner face.

Lemma 2 (Push-through lemma)**.** *For n primes P_i*

$$P_i \quad : \quad \langle m_i, \vec{X}_i, \{\vec{X}_i\}\rangle \to \langle 1, (Y_i^{\mathrm{B}}), Y_i\rangle,$$

and permutation π, there exists a permutation $\overline{\pi}_{\vec{m},\vec{X}}$, which depends solely on π, $\vec{m}$, and $\vec{X} = (\vec{X}_0, \ldots, \vec{X}_{n-1})$, s.t.,

$$\vdash \pi \circ (P_0 \otimes \cdots \otimes P_{n-1}) = (P_{\pi(0)} \otimes \cdots \otimes P_{\pi(n-1)}) \circ \overline{\pi}_{\vec{m},\vec{X}}.$$

4.2 ***Place$_{\mathbf{L}}$*** *expressions*

Let **Place$_{\mathbf{L}}$** expressions be all expressions in the term language generated by $\circ$, and $\otimes$ from *bmerge*$_{m,\vec{X}}$ (defined below) and $\gamma_{I,J}$. Thus, **Place$_{\mathbf{L}}$** consists of all expressions denoting place graphs paired with identities on local names. We shall start by proving that the theory is complete for **Place$_{\mathbf{L}}$** expressions.

To that end, we extend the place merging expression *join* to local interfaces.

Definition 4 (binding join)**.** *For sets of names X and Y let $bjoin_{(X,Y)}$, the* binding *join bigraph, be defined as*

$$bjoin_{(X,Y)} \quad \stackrel{\text{def}}{=} \quad (X \uplus Y)((join \otimes \mathsf{id}_{X \uplus Y}) \circ (\ulcorner X \urcorner \otimes \ulcorner Y \urcorner)).$$

We also define an iterated version

Categorical axioms				
(C1)	$A\ \mathsf{id}_I$	$=$	$A = \mathsf{id}_J\ A$	$(A : I \to J)$
(C2)	$A(BC)$	$=$	$(AB)C$	
(C3)	$A \otimes \mathsf{id}_\epsilon$	$=$	$A = \mathsf{id}_\epsilon \otimes A$	
(C4)	$A \otimes (B \otimes C)$	$=$	$(A \otimes B) \otimes C$	
(C5)	$\mathsf{id}_I \otimes \mathsf{id}_J$	$=$	$\mathsf{id}_{I \otimes J}$	
(C6)	$(A_1 \otimes B_1)(A_0 \otimes B_0)$	$=$	$(A_1 \circ A_0) \otimes (B_1 \circ B_0)$	
(C7)	$\gamma_{I,\epsilon}$	$=$	id_I	
(C8)	$\gamma_{J,I}\ \gamma_{I,J}$	$=$	$\mathsf{id}_{I \otimes J}$	
(C9)	$\gamma_{I,K}(A \otimes B)$	$=$	$(B \otimes A)\gamma_{H,J}$	$(A : H \to I, B : J \to K)$
Link axioms				
(L1)	x/x	$=$	id_x	
(L2)	$/y \circ y/x$	$=$	$/x$	
(L3)	$/y \circ y$	$=$	id_ϵ	
(L4)	$z/(Y \uplus y)(\mathsf{id}_Y \otimes y/X)$	$=$	$z/(Y \uplus X)$	
Place axioms				
(P1)	$join(1 \otimes \mathsf{id}_1)$	$=$	id_1	
(P2)	$join(join \otimes \mathsf{id}_1)$	$=$	$join(\mathsf{id}_1 \otimes join)$	
(P3)	$join\ \gamma_{1,1,(\emptyset,\emptyset)}$	$=$	$join$	
Binding axioms				
(B1)	$(\emptyset)P$	$=$	P	
(B2)	$(Y)\ulcorner Y \urcorner$	$=$	$\mathsf{id}_{(Y)}$	
(B3)	$(\ulcorner X \urcorner^Z \otimes \mathsf{id}_Y)(X)P$	$=$	P	$(P : I \to \langle 1, (Z), Z \uplus X \uplus Y \rangle$
(B4)	$(((Y)(P)) \otimes \mathsf{id}_X)G$	$=$	$(Y)(P \otimes \mathsf{id}_X)G$	
(B5)	$(X \uplus Y)P$	$=$	$(X)((Y)P)$	
Ion axioms				
(N1)	$(\mathsf{id}_1 \otimes \alpha)K_{\vec{y}(\vec{X})}$	$=$	$K_{\alpha(\vec{y})(\vec{X})}$	
(N2)	$K_{\vec{y}(\vec{X})}\sigma^{\mathbf{loc}}$	$=$	$K_{\vec{y}((\sigma^{\mathbf{loc}})^{-1}(\vec{X}))}$	

TABLE I: Axioms for binding bigraphs

Definition 5 (binding merge)**.** *For all $m \geq 0$ we define $bmerge_{m,\vec{X}}$ recursively, by*

$$\begin{aligned} bmerge_{0,()} &\stackrel{\text{def}}{=} 1 \\ bmerge_{m+1,\vec{X}Y} &\stackrel{\text{def}}{=} bjoin_{(\{\vec{X}\},Y)} \circ (bmerge_{m,\vec{X}} \otimes \mathsf{id}_Y) \end{aligned}$$

Binding join and merge behave similarly as their underlying place expressions when composed with permutations or themselves (refer the place graph axioms of Table I), though, as they have (local) names on their faces their interplay with names is not as simple. The lemma below reflects this, and also states that merging a product of binding merges yields a binding merge.

Lemma 3.

$$\begin{aligned} &\vdash bjoin_{(X_1,X_0)} \circ \gamma_{1,1,(X_0,X_1)} = bjoin_{(X_0,X_1)}, \\ &\vdash bmerge_{m,\pi(\vec{X})} \circ \pi = bmerge_{m,\vec{X}}, \\ &\vdash bmerge_{k,\vec{X}} \circ (\textstyle\bigotimes_{i<k} bmerge_{m_i,\vec{X}_i}) = bmerge_{m,\vec{X}}, \end{aligned}$$

where in the last equation $m = \sum_{i<k} m_i$ and
$\vec{X} = \vec{X_0} \dots \vec{X_{k-1}}$.

Using binding merge, we define and prove sufficient a normal form for $\mathbf{Place_L}$ expressions.

Lemma 4 (Normal form for $\mathbf{Place_L}$ expressions)**.** *For every*
$\mathbf{Place_L}$ expression E

$$\vdash E \;=\; (bmerge_{m_0,\vec{X_0}} \otimes \cdots \otimes bmerge_{m_{k-1},\vec{X_{k-1}}}) \circ \pi$$

for some $k \geq 0$ and permutation expression π s.t. the composition is well defined.

With the help of Lemma 3 the proof is simple by induction on the structure of expressions.

Note that in a strict symmetric monoidal category the categorical axioms are known to be complete for $\circ$ and $\otimes$ of the symmetries $\gamma_{I,J}$ — hence the theory is complete for permutations.

Full completeness for $\mathbf{Place_L}$ expressions follows with the help of the uniqueness properties stated in Theorem 1. These yield a number of equations which are provable within the theory.

Proposition 4 (Completeness for $\mathbf{Place_L}$ expressions)**.** *If*
$\vdash E = \bigotimes_{i<k} bmerge_{m_i,\vec{X_i}} \circ \pi$ *and* $\vdash F = \bigotimes_{j<l} bmerge_{n_j,\vec{Y_j}} \circ \pi'$ *and* $\models E = F$,
then $\vdash E = F$.

4.3 $\textbf{Link}_{\mathbf{G}}$ *expressions*

We now consider the class of global link expressions, those bigraph expressions generated by composition and tensor of closure and substitution. We will refer to this collection of expressions as $\textbf{Link}_{\mathbf{G}}$. Our term language for binding bigraphs has the same constructs for linking as the language used by Milner for pure bigraphs [12]. Since we also have the exact same axioms for global link expressions, it is easily seen that the proof that the axiomatic theory for the binding bigraph term language is complete for global link expressions is entirely the same.

Proposition 5 (Link completeness)**.** *The theory is complete for link expressions.*

4.4 Linear bigraph expressions

We now define an important kind bigraph expressions – **linear** expressions, which we shall prove to be a syntactic analogue to name-discrete bigraphs, in the sense that any name-discrete bigraph has a linear expression.

Definition 6 (Linearity)**.** *A binding bigraph expression is* ***linear*** *iff it contains only linkings of the form y/x.*

In other words, in linear expressions all substitutions are renamings, and there are no closures. This is an inductive property with respect to the term language, which we will utilize to full effect in the following sections.

We start by establishing some basic properties of linear expressions. The proofs of the following lemmas are all by induction on the structure of expressions.

Lemma 5. *If E is linear expression, then $\vdash E = E' \otimes \alpha$, where E' is linear and has local innerface.*

Lemma 6. *If $E : \langle m, \vec{U}, \{\vec{U}\}\rangle \to \langle n, \vec{Y}, \{\vec{Y}\} \uplus V\rangle$ is a linear expression with local innerface, then*

$$\vdash E \ \circ \ \bigotimes_{i<m}(\vec{u_i})/(\vec{Z_i}) = \left(\left(\bigotimes_{i<n}(\vec{y_i})/(\vec{X_i})\right) \ \otimes \ \mathsf{id}_V\right) \ \circ \ E',$$

for some $\vec{y}$, $\vec{X}$, and E' where E' is linear with local innerface.

We shall use the following proposition to show completeness for ion-free expressions in the following section. Importantly, it also constitutes a step towards a syntactic normal form for bigraph expressions, analogous to the semantic normal form we established in Theorem 1, item 4.

Proposition 6 (Underlying linear expression)**.** *For any expression G denoting a bigraph of outer width n, there exists a wiring ω, a linear expression E, and a local renaming $\bigotimes_{i<n}(\vec{y_i})/(\vec{X_i})$, s.t.,*

$$\vdash G = (\bigotimes_{i<n}(\vec{y_i})/(\vec{X_i}) \otimes \omega) \circ E.$$

The proof is by structural induction on G, using the lemmas above [5].

4.5 Ion-free expressions

Let us now consider ion-free expressions – all expressions in our term language, that does not contain ions ($K_{\vec{y}(\vec{X})}$). We proceed as above, by showing that ion-free expressions can be decomposed into simpler expressions.

Lemma 7. *If $E = E_1 \circ E_2$ or $E = E_1 \otimes E_2$ is linear, ion-free, and with local inner and outer face, then E_1 and E_2 are also linear and ion-free with local inner and outer face.*

Lemma 8. *If E is linear and ion-free of width n with local inner and outer face, then $\vdash E = \bigotimes_{i<n}(\vec{y_i})/(\vec{x_i}) \circ G^P$, where $G^P \in \boldsymbol{Place_L}$.*

Lemma 9. *If E is linear and ion-free, then there exists concretions, E', and a renaming α s.t. $\vdash E = (\bigotimes_{i<n} \ulcorner X_i \urcorner^{Z_i} \circ E') \otimes \alpha$, with E' linear and ion-free and local inner and outer face.*

With the help of the above lemmas we can now establish a normal form for ion-free expressions.

Lemma 10 (A normal form for ion-free expresssions)**.** *For all ion-free expressions G of width n*

$$\vdash G = \omega \otimes \left(\bigotimes_{i<n} (Y_i) \left((\rho \otimes \mathsf{id}_1) \circ \ulcorner X_i \urcorner \right) \right) \circ G^P.$$

where $G^P \in \boldsymbol{Place_L}$.

Completeness for ion-free expressions follows easily.

Corollary 1 (The theory is complete for ion-free expressions)**.**

4.6 Syntactic Normal Form

Corresponding to the four classes of normal forms in Theorem 1 we define four classes of syntactic normal forms for binding bigraph expressions:

Definition 7 (Syntactic binding discrete normal form (BDNF))**.**

$$\begin{array}{lll}
\text{MDNF} \quad M & ::= & (K_{\vec{y}(\vec{X})} \otimes \mathsf{id}_Z)P \\
\text{PDNF} \quad P & ::= & (X)\left(merge_{n+k} \otimes \mathsf{id}_Y\right) \\
 & & \quad (\ulcorner \alpha_0 \urcorner \otimes \cdots \otimes \ulcorner \alpha_{n-1} \urcorner \otimes M_0 \otimes \cdots \otimes M_{k-1})\,\pi \\
\text{DDNF} \quad D & ::= & (P_0 \otimes \cdots \otimes P_{n-1})\pi \otimes \alpha \\
\text{BDNF} \quad B & ::= & (\bigotimes_{i<n}(\vec{y_i})/(\vec{X_i}) \otimes \omega)D.
\end{array}$$

The proofs of the following lemmas go by induction on the number of ions. As we have established completeness for ion-free expressions, we have the base case.

Lemma 11 (All BDNF forms are closed under composition with isos)**.**

We also need that DDNF expressions are closed under composition.

Lemma 12 (DDNF is closed under composition)**.** *For all composable* DDNF*s* C, D*, there exists a* DDNF D'*, s.t.* $\vdash D \circ C = D'$.

We now state the correspondence between our semantic normal form (Theorem 1) and the syntactic normal form above. Moreover, we state that linearity is, in fact, a syntactic correspondent to name-discreteness (item 3 in the following proposition):

Proposition 7 (provable normal forms)**.** *Let* E *be a linear expression, and* G *any expression.*

(1) If E *denotes a free discrete molecule, then* $\vdash E = M$ *for some* MDNF*.*
(2) If E *denotes a name-discrete prime, then* $\vdash E = P$ *for some* PDNF P*.*
(3) $\vdash E = D$ *for some* DDNF D*.*
(4) $\vdash G = B$ *for some* BDNF B*.*

We are now able to state the formal completeness proposition, using our results for linear expressions to bridge the gap to the full binding bigraph term language.

As we have laboured to establish a correspondence between each level of BDNF form and each level of the semantic normal form, in the proofs we are able to proceed by case analysis on the form of the bigraph the expression denotes, and then apply the uniqueness properties spelled out in Theorem 1 to yield a number of equations that are provable within our theory. We refer to the companion technical report [5] for more details on the proofs.

Proposition 8 (Linear completeness)**.** *If* E *and* E' *are linear expressions and* $E = E'$*, then* $\vdash E = E'$*.*

Theorem 2 (Soundness and Completeness)**.** *For all binding bigraph expressions* E *and* F*,* $\vDash E = F$ *iff* $\vdash E = F$*.*

5. Term Language and Normal Forms – by Example

We shall use our examples from Section 2 to give a few examples of the term language and the syntactic binding discrete normal form.

Using a modicum amount of shorthand, an expression for C is $((x)(\text{secret}_x \circ\ 1)) \otimes \text{pda}_z \circ\ 1$, while B can be expressed for example as

$$\begin{aligned}&(\mathsf{id}_2 \otimes /\{e0, e1\}) \circ (\ (\text{server}_{e0(\{x\})})\ \otimes \\ &\quad (\mathsf{id}_{\langle 1,e1\rangle} \otimes /\{f0, f1\})(\text{office} \circ 1 \otimes \mathsf{id}_{\{e1,f0,f1\}}) \\ &\quad (\mathit{merge}_3 \otimes \mathsf{id}_{\{e1,f0,f1\}})(\text{pc}_{e1} \circ 1 \otimes \text{pda}_{f0} \circ 1 \otimes \text{pda}_{f1} \circ 1))).\end{aligned}$$

Hence, A can be expressed, simply by putting a $\circ$ between the two expressions for B and C.

On the other hand, giving an expression on BDNF for either bigraph, requires us to break it down into molecules, prime parts and not use non-linear linkage except at the topmost level. As an example, we give an expression on normal form for B:

$$\begin{aligned}
&(\mathsf{id}_2 \otimes /\{e0, e1\} \otimes /\{f0, f1\}) \circ \\
&\quad(\ \ (\emptyset)(\mathit{merge}_1 \otimes \mathsf{id}_{e0})(\mathrm{server}_{e0(\{x\})} \circ (x)(\mathit{merge}_1 \otimes \mathsf{id}_x)(\ulcorner x \urcorner)) \otimes \\
&\qquad (\emptyset)(\mathit{merge}_1 \otimes \mathsf{id}_{\{e1,f0,f1\}})(\mathrm{office} \circ \\
&\qquad\quad (\emptyset)(\mathit{merge}_3 \otimes \mathsf{id}_{\{e1,f0,f1\}})(\mathrm{pc}_{e1} \circ 1 \otimes \mathrm{pda}_{f0} \circ 1 \otimes \mathrm{pda}_{f1} \circ 1)) \\
&\quad \otimes\ \mathsf{id}_\epsilon).
\end{aligned}$$

Here we do not show identity permutations, and we write 1 instead of PDNF for 1 (which is $(\emptyset)(\mathit{merge}_0 \otimes \mathsf{id}_\epsilon)$).

6. Related and further work

Bigraphical reactive systems are related to graph transformation systems using the double pushout construction [7] and, recently, it has also been investigated how to derive bisimulation congruences in the double pushout approach to graph rewriting [8].

Recent work on spatial logics [4] for pure bigraphs utilizes the axiomatization of pure bigraphs by Milner [12]. An obvious line of further work is to utilize the algebraic theory presented here for binding bigraphs to extend the spatial logics to binding.

As mentioned in the introduction, jointly with the other members of our Bigraphical Programming Languages group, we are are currently working on an implementation of bigraphical reactive systems.

Further work is needed to relate tools based on graph rewriting to our work on Bigraphical Programming Languages.

Currently our experimental implementation of bigraphical reactive systems represents bigraphs internally by *normal form bigraphical expressions* that denote bigraphs. We have also developed a proposal for a surface language which users can use to define bigraphical reactive systems — expressions of the surface language denote binding bigraphs and can thus be transformed to binding discrete normal forms: the proofs of the normal form theorems of this paper are constructive in nature and thus define algorithms than can be used to transform arbitrary bigraph expressions into normal form.

The core problem of implementing the dynamics of bigraphical reactive systems is the *matching problem*, that is, to determine for a given bigraph and reaction rule whether and how the reaction rule can be applied to rewrite the bigraph.

The abstract semantic definition of matching, as defined in the theory of bigraphs [9], is roughly as follows (omitting many details): Given a reaction rule with redex R and reactum R' (with R and R' both bigraphs), and a bigraph A (the agent to be rewritten), if $A = C \circ R \circ d$, then it can be rewritten to $C \circ R' \circ d$. Here $\circ$ denotes composition of bigraphs. In other words, if the reaction rule *matches* A, in the sense that A can be decomposed into a context C, redex R and a parameter d, then A can be rewritten.

Phrased in terms of binding bigraph *expressions*, the decision problem for matching is then roughly the following. Given binding bigraph expressions R, A, C, and d, determine whether $\vDash A = C \circ R \circ d$ holds. We have worked out an *inductive characterization* of when $\vDash A = C \circ R \circ d$ holds, by induction on the normal forms for A and R (the input to a matching algorithm). It is a precise characterization in the sense that it is both sound and complete. This provides a detailed analysis of the matching problem, and paves the way for developing and proving correct an actual matching algorithm (which, given A and R, must find a C and d such that $\vDash A = C \circ R \circ d$ holds). We will report on our work on the inductive characterization and on an actual matching algorithm in a subsequent paper.

We intend to use the implementation of bigraphical reactive systems to evaluate also in practice how well bigraphical models of ubiquitous systems [3] work.

Acknowledgements

We are grateful for useful discussions of this work with all members of the BPL group at the IT University of Copenhagen, in particular Arne Glenstrup and Søren Debois; and with Robin Milner.

Appendix A. Definition of Binding Bigraphs

We recall the definition of binding bigraphs [9].

Definition 8 (binding signature)**.** *A* ***binding signature*** $\mathcal{K}$ *is a set of* ***controls****. For each* $K \in \mathcal{K}$ *it provides a pair of finite ordinals: the* ***binding arity*** $\mathrm{ar}_{\mathbf{b}}(K) = h$ *and the* ***free arity*** $\mathrm{ar}_{\mathbf{f}}(K) = k$*. We write* $\mathrm{ar}(K) = \mathrm{ar}_{\mathbf{b}}(K) + \mathrm{ar}_{\mathbf{f}}(K)$*.*

Definition 9 (binding interface)**.** *A* ***binding interface*** $I = \langle m, loc, X \rangle$*, consists of a* ***width*** m*, a finite set of* ***names*** X*, and a* ***locality map*** $loc : X \to m \uplus \bot$*, which associates some of the names in* X *with a location in* m*; if* $loc(x) = i \in m$*, we say* x *is* ***located*** *at* i *or* ***local*** *to* i*. When* $loc(x) = \bot$ *we say* x *is* ***global****.*

For an interface $I = \langle m, loc, X \rangle$ we shall typically represent the locality map by a vector of disjoint subsets $\vec{X} = (X_0, \ldots, X_{m-1})$, where X_i is the set of names local to $i \in m$. If I is global, meaning that all names in I are global, then we may write I simply as $\langle m, X \rangle$; just m, if $X = \emptyset$; or just X, if $m = 0$.

We call I **prime** if $m = 1$. In that case, we shall sometimes write I as $\langle (X), Y \rangle$; just (X), if it is local; or just $\langle Y \rangle$, if it is global.

We use ϵ to denote the interface $\langle 0, (), \emptyset \rangle$.

A binding bigraph will have two binding interfaces and will be a pairing of a **place graph**, and a **link graph** following a structural requirement, the **scope rule** (see Definition 13).

We start by calling to mind the definitions of place graphs and link graphs.

Definition 10 (place graph)**.** *A (concrete)* ***place graph*** *over signature* $\mathcal{K}$ $G = (V, ctrl, prnt) : m \to n$ *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 parent map *prnt* represents a forest of n unordered trees. The widths m and n of $G : m \to n$ index G's **sites** $0, \ldots, m-1$ and **roots** $0, \ldots, n-1$, respectively. We use ϵ to denote the width 0. A place graph with inner width 0 is called an **agent**.

Place graphs are composed as follows. Let $G_i = (V_i, ctrl_i, prnt_i) : m_i \to m_{i+1}$ ($i \in \{0, 1\}$) 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 $\mathsf{id}_m \stackrel{\text{def}}{=} (\emptyset, \emptyset, \mathsf{id}_m) : m \to m$.

The tensor product $I \otimes J$ of two interfaces $I = m$ and $J = n$ is simply $m + n$, and the tensor product of two place graphs $F : k \to l$ and $G : m \to n$ with disjoint node sets is $F \otimes G : k + m \to l + n$. It consists of placing the two forests side-by-side (see [9, Definition 7.5] for a formal definition). Note that $\mathsf{id}_\epsilon = \mathsf{id}_0$ is the unit for $\otimes$, in the sense that $F \otimes \mathsf{id}_\epsilon = \mathsf{id}_\epsilon \otimes F = F$, for all place graphs F. Thus, an iterated tensor product $F_0 \otimes \cdots \otimes F_{k-1}$ equals id_ϵ in case $k = 0$.

Two concrete place graphs G_0 and G_1 are said to be **support equivalent**, $G_0 \simeq G_1$, if they differ only by a bijection between their node sets. An **abstract place graph** is an $\simeq$-equivalence class of concrete place graphs. Composition and identity of abstract place graphs is given by composition and identity of concrete place graphs, and this provides a well-defined **category of place graphs** with interfaces as objects and abstract place graphs as morphisms. The induced tensor product on abstract place graphs, defined by $[F]_\simeq \otimes [G]_\simeq \stackrel{\text{def}}{=} [F \otimes G]_\simeq$, makes it into a strict symmetric monoidal category.

Definition 11 (link graph)**.** *A (concrete)* ***link graph*** G *over a signature* $\mathcal{K}$*, is a tuple* $(V, E, ctrl, link) : X \to Y$ *with finite sets of nodes* V*, edges* E*,* ***inner names*** X*, and* ***outer names*** Y*. As place graphs it has a control map* $ctrl : V \to \mathcal{K}$*. The function* $link : X \uplus P \to E \uplus Y$ *maps* ***points****, i.e., inner names* X *and ports* $P = \sum_{v \in V} \mathrm{ar}(ctrl\, V)$ *of* G *to* ***links****, i.e., outer names* Y *and edges* E*.*

We call a link **idle** if it has no preimage under *link*. An outer name is an **open** link, and an edge is a **closed** link. A point is called **open** if its link is open, otherwise closed. Further, we call two distinct points on the same link **peers**.

The composition of two link graphs $G_i = (V_i, E_i, ctrl_i, link_i) : X_i \to X_{i+1}$ ($i \in \{0, 1\}$) is defined when $V_0 \cap V_1 = \emptyset$ and $E_0 \cap E_1 = \emptyset$; and is then $G_1 \circ G_0 \stackrel{\text{def}}{=} (V, E, ctrl, link) : X_0 \to X_2$; where $V = V_0 \uplus V_1$, $E = E_0 \uplus E_1$, $ctrl = ctrl_0 \uplus ctrl_1$, and $link = (\mathsf{id}_{E_0} \uplus link_1) \circ (link_0 \uplus \mathsf{id}_{P_1})$.

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

The tensor product of two link graph interfaces X and Y is the disjoint union, $X \uplus Y$, and is defined only when X and Y are disjoint. Tensor product of link graphs $G_i = (V_i, E_i, ctrl_i, link_i) : X_i \to Y_i$ is the disjoint union of the underlying constituents $G_0 \otimes G_1 \stackrel{\text{def}}{=} (V_0 \uplus V_1, E_0 \uplus E_1, ctrl_0 \uplus ctrl_1,$ $link_0 \uplus link_1) : X_0 \otimes X_1 \to Y_0 \otimes Y_1$, and is defined only when the interfaces are defined.

Definition 12 (binding bigraph)**.** *A (**concrete**) **binding bigraph** $G = (V, E, ctrl, G^{\mathbf{P}}, G^{\mathbf{L}}) : I \to J$ over a signature $\mathcal{K}$ has an **inner interface** (or **inner face**) $I = \langle m, loc_I, X\rangle$ and an **outer interface** (or **outer face**) $J = \langle n, loc_J, Y\rangle$. Here V, E and $ctrl$ are finite sets of nodes, edges, and a control map $ctrl : V \to \mathcal{K}$, exactly as for link graphs.*

The fourth component $G^{\mathbf{P}} = (V, ctrl, prnt) : m \to n$ is a place graph, while the fifth $G^{\mathbf{L}} = (V, E, ctrl, link) : X \to Y$ is a link graph.

*We require that G adheres to the **scope rule** below.*

Definition 13 (scope rule)**.** *Let the **binders** of G be the binding ports of nodes in V and the local names of its outer face J.*

If p is a binder located at a node or root w, then for all peers p' of p, $loc(p') = w'$ must imply $w' = prnt^k_{G^{\mathbf{P}}}(w)$, for some $k > 0$.

We say that a link is **bound** if it contains a binder, otherwise **free**. As usual, we extend this terminology to the points in the link. A binding bigraph $G : I \to J$ is said to be **free** if its outer face J is global, i.e., the image of loc_J is $\perp$.

A binding bigraph G is given by its underlying place $G^{\mathbf{P}}$ and link graph $G^{\mathbf{L}}$ and its binding interfaces I and J. We write $G = \langle G^{\mathbf{P}}, G^{\mathbf{L}}\rangle : I \to J$. We shall sometimes use a variant of the 5-tuple notation where we inline the components unique to the place graph and link graph components, i.e., $G = (V, E, ctrl, prnt, link) : I \to J$.

We define a notation for the underlying set of vectors of names: Given a vector of disjoint name sets $\vec{Y}$, $\{\vec{Y}\}$ denotes the disjoint union of the sets in the vector. Composition and tensor product of concrete binding bigraphs $G_i = \langle G^{\mathbf{P}}_i, G^{\mathbf{L}}_i\rangle : I_i \to J_i$ are given by composition and tensor product of their underlying place and link graphs, and by the tensor product of binding interfaces. We have only to explain the latter: Tensor product of binding interfaces $I_i = \langle m_i, \vec{X}_i, X_i\rangle$ is $I_0 \otimes I_1 \stackrel{\text{def}}{=} \langle m_0 + m_1, \vec{X}_0\vec{X}_1, \{\vec{X}_0\} \uplus \{\vec{X}_1\}\rangle$ (letting juxtaposition denote vector concatenation), and is defined when the name sets are disjoint. Hence, if the bigraphs above have disjoint node and edge sets, $G_1 \circ G_0 \stackrel{\text{def}}{=} \langle G^{\mathbf{P}}_1 \circ G^{\mathbf{P}}_0, G^{\mathbf{L}}_1 \circ G^{\mathbf{L}}_0\rangle : I_0 \to J_1$ is defined if $I_1 = J_0$;

and $G_1 \otimes G_0 \stackrel{\text{def}}{=} \langle G_1^{\mathbf{P}} \otimes G_0^{\mathbf{P}}, G_1^{\mathbf{L}} \otimes G_0^{\mathbf{L}} \rangle \ : \ I_0 \otimes I_1 \ \rightarrow \ J_0 \otimes J_1$ if the tensor products of the interfaces are defined. (See [9, Chapter 11] for more details.)

The identity for composition is given by a pairing of the identities for composition for place graphs and link graphs. If $I = \langle m, loc, X \rangle$ then $\mathsf{id}_I \stackrel{\text{def}}{=} \langle \mathsf{id}_m, \mathsf{id}_X \rangle : I \rightarrow I$.

We shall use the following notation for iterated tensor product: $\bigotimes_{i<n} P_i = P_0 \otimes P_1 \otimes \cdots \otimes P_{n-1}$. The identity for tensor is id_ϵ; thus, an iterated tensor product $P_0 \otimes \ldots \otimes P_{n-1}$ equals id_ϵ in case $n = 0$. Composition binds tighter than tensor product, and abstraction $(Y)P$ and $\bigotimes$ binds as far right as possible.

We say that two concrete binding bigraphs G_0 and G_1 are **lean-support equivalent**, denoted $G_0 \mathrel{\ddot{\simeq}} G_1$ iff they differ only by a bijection between their nodes and their non-idle edges; idle edges are disregarded entirely.

Abstract binding bigraphs are $\ddot{\simeq}$-equivalence classes of concrete binding bigraphs. Composition, tensor and identity of abstract binding bigraphs are given by composition, tensor and identity of the underlying concrete bigraphs. Taking interfaces as objects and abstract binding bigraphs as morphisms we have a **category of binding bigraphs**. Finally, a **ground** bigraph is a bigraph with inner face ϵ. We shall also refer to such a bigraph as an **agent**. A bigraph $G : I \rightarrow J$ is called **prime**, if I is local and J is prime.

We shall need to consider and distinguish several forms of **discreteness**, which we define below.

Definition 14 (Variants of discreteness)**.**

- *We say that a bigraph is* ***discrete*** *iff every free link is an outer name and has exactly one point.*
- *A bigraph is* ***name discrete*** *iff it is discrete and every bound link is either an edge, or (if it is an outer name) has exactly one point.*

Note that name-discrete implies discrete. Name-discreteness is defined to impose exactly the same level of constraints on local and global linkage upon names. We utilize this in the normal form we define. Discreteness and name-discreteness share several nice properties.

Lemma 13. *If A and B are discrete, then $A \otimes B$, $(Y)A$, and AB are also discrete. The same holds for name-discrete bigraphs A and B.*

References

[1] Birkedal, L., Debois, S., Elsborg, E., Hildebrandt, T., and Niss, H. 2006. Bigraphical Models of Context-aware Systems. In *In Proceedings of FOSSACS 2006*.

[2] Birkedal, Lars. 2004. Bigraphical Programming Languages—A LaCoMoCo Research Project. In *Second UK UbiNet Workshop, Cambridge*.

[3] Birkedal, Lars, Debois, Søren, Elsborg, Ebbe, Hildebrandt, Thomas, and Niss, Henning. 2005. Bigraphical Models of Context-aware Systems. Tech. Report TR–2005–74, IT University of Copenhagen, Rued Langgards Vej 7, DK-2300 Copenhagen V.

[4] Conforti, G., Macedonio, D., and Sassone, V. 2005. Spatial Logics for Bigraphs. In *Proc. of International Colloquium on Automata, Languages and Programming*.

[5] Damgaard, T.C. and Birkedal, L. 2005. Axiomatization of Binding Bigraphs (Revised). Tech. Report TR–2005–71, IT University of Copenhagen.

[6] Debois, S. and Damgaard, T. C. 2005. Bigraphs by Example. Tech. Report TR–2005–61, The IT University of Copenhagen.

[7] Ehrig, H. 1979. Introduction to the theory of graph grammars. In *Graph Grammars and their application to Computer Science and Biology*, Number 73 in Lecture Notes in Computer Science. Springer-Verlag, 1–69.

[8] Ehrig, H. and König, B. 2004. Deriving Bisimulation congruences in the DPO approach to graph rewriting. In *Foundations of Software Science and Computation Structures*, Volume 2987 of *Lecture Notes in Computer Science*. Springer-Verlag.

[9] Høgh Jensen, Ole and Milner, Robin. 2004. Bigraphs and mobile processes (revised). Tech. Report 580, University of Cambridge.

[10] Jensen, O.H. 2006. *Mobile Processes in Bigraphs (Forthcoming)*. PhD thesis, Univ. of Aalborg.

[11] Leifer, James J. and Milner, Robin. 2004. Transition systems, link graphs and Petri nets. Tech. Report 598, University of Cambridge.

[12] Milner, Robin. 2005. Axioms for bigraphical structure. *Mathematical Structures in Computer Science 15*, 06 (December), 1005–1032.

[13] Milner, Robin. 2005. Pure bigraphs. Tech. Report 614, University of Cambridge.