

Typed polyadic pi-calculus in bigraphs *

Mikkel Bundgaard
IT University of Copenhagen

Vladimiro Sassone
ECS, University of Southampton

Abstract

Bigraphs have been introduced with the aim to provide a topographical meta-model for mobile, distributed agents that can manipulate their own communication links and nested locations. In this paper we examine a presentation of type systems on bigraphical systems using the notion of sorting. We focus our attention on the *typed polyadic π -calculus* with capability types à la Pierce and Sangiorgi, which we represent using a novel kind of link sorting called *subsorting*. Using the theory of *relative pushouts* we derive a labelled transition system which yield a coinductive characterisation of a behavioural congruence for the calculus. The results obtained in this paper constitute a promising foundation for the presentation of various type systems for the (polyadic) π -calculus as sortings in the setting of bigraphs.

Categories and Subject Descriptors F.3.2 [Logics and Meanings of Programs]: Semantics of Programming Languages—Process models

General Terms Foundation of computation, semantics of concurrency, foundations of distributed and mobile systems.

Keywords Bigraphs, typed polyadic π -calculus, sortings, subsorting, bisimulation congruences, relative pushouts.

Introduction

Bigraphical reactive systems (BRS) [8] have been proposed as a topographical meta-model for mobile, distributed agents that can manipulate their own communication linkage and nested locations. Bigraphs generalise both the link structure characteristic of the π -calculus and the nested location structure characteristic of the calculus of Mobile Ambients. A bigraph consists of two overlapping structures: a *place graph* and a *link graph*. The place graph is a tuple of unordered trees that represents the topology of the system. Its *roots* contain nodes which represent locations or process constructors. Some of the leaves may be *sites* to be filled by other bigraphs, so giving rise to bigraphical (multi-hole) contexts. Each node is typed with a *control* which prescribes its number of *ports*. The link graph represents the system's connectivity. It links together ports and names in the bigraph's inner and outer interfaces. Names in

the inner interface represent connection points offered to bigraphs that may fill sites; those in the outer interface represent free names exported by the system.

Binding bigraphs extend this basic structure – known as pure bigraphs – by allowing some of the ports of a node to be ‘binding,’ meaning that all other points linked to the port must lie inside the node. A binding port enforces a notion of scope on a bigraph's links, resembling in such a way the usual notion of binders in the λ - and the π -calculus. Binding interfaces record topological information (viz., sites and roots), inner and outer namesets, as well as the binding of names to locations. Fig. 1 depicts a binding bigraph with *inner interface* $\langle 3, (\{x_2\}, \emptyset, \emptyset), \{x_0, x_1, x_2\} \rangle$, reflecting that it consists of three sites (shaded in the picture) only the first of which contains a local name, the binder x_2 . The bigraph's *outer interface* is $\langle 2, (\emptyset, \emptyset), \{y_0, y_1, y_2\} \rangle$, with two roots, or locations (drawn in dashed lines), and only global names.

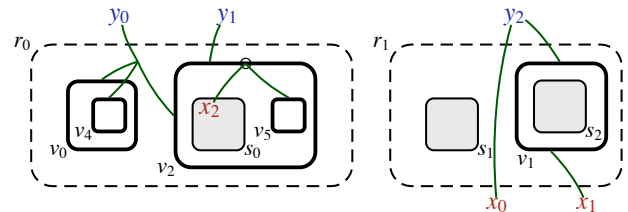


Figure 1. A binding bigraph

Often when representing systems and calculi as bigraphical reactive systems one needs to constrain the allowable compositions of nodes and links. Examples of such constraints are Jensen's representation of the π -calculus with guarded sum [6], where for instance nodes of a given control **sum** must not contain nodes of the same sort as immediate children, or Leifer and Milner's treatment of Petri nets [13], where transitions can only be connected to places and vice versa. A *sorting* is used to enforce constraints such as these on a class of bigraphs.

The *polyadic π -calculus* [14] is a generalisation of the monadic π -calculus, whereby a single message can carry a tuple of names rather than a single one. This has the immediate consequence that communication can go ‘wrong’ in that communicating parties may not agree on the number of names exchanged in a communication. A type system is needed to ensure that only well-formed processes are allowed by the formalism. In his original presentation of the polyadic π -calculus in [14], Milner introduced a simple sorting discipline to ensure ‘arity’ safety of communications. Pierce and Sangiorgi presented in [17] a generalisation of Milner's sorting with *capability types* and a structural subtype relation on sorts, which in addition can ensure that well-typed processes use names only for input (resp. output) actions, according to a predefined discipline. Inspired by this work and following Jensen and Milner's encoding of the π -calculus in binding bigraphs [8], we begin the study an elementary theory of link-subtyping for bigraphical systems.

* Supported by ‘DisCo: Semantic Foundations of Distributed Computation,’ EU IHP ‘Marie Curie’ HPMT-CT-2001-00290.

Related work. Place sortings enforce a discipline on the allowed nesting of nodes in place graphs, and was introduced in [15] to represent guarded sums using a two-sorted syntax in finite pure CCS. Place sortings have also been examined in [16], where the author puts forward the notion of *kind* bigraphs, i.e., bigraphs where the notion of ‘atomic’/‘non-atomic’ controls is generalised in order to specify which kind of controls can be contained inside each other. In his thesis [6], Jensen utilises place sortings and puts their foundation of a solid footing by means of general requirements on the sorting functors which ensure that they respect the behavioural theory of the unsorted model. In [1] Birkedal *et al.* investigate bigraphical reactive systems as models for global ubiquitous computing. A place sorting that partitions the set of controls in the system is deployed to obtain the results in that paper.

Link sortings enforce requirements on the kind of linkage that can occur in a bigraph. Leifer and Milner [13] introduce link sortings to encode condition-event Petri nets; the sorting ensures well-formed bigraphs indeed represent Petri nets.

Sewell [20] generalised the capability types of Pierce and Sangiorgi to also contain information about locality, so that one can distinguish between local, global communication, or no communication at all. In [5, 4] Hennessy *et al.* investigate a distributed π -calculus together with a type system which generalises Pierce and Sangiorgi’s by ensuring that processes only can access resources they are given permission to. Contrary to Pierce and Sangiorgi’s, the subtype relation in *loc. cit.* has bounded meets, a property which is essential to establish our results. Deng and Sangiorgi [3] use similar capability types and subtyping in order to study the effect of types on the algebraic theory of the (monadic) π -calculus.

König [12] presents a graphical approach to the asynchronous polyadic π -calculus using a hypergraph rewriting semantics in the double-pushout approach. The representation of prefixes differs crucially from ours: whilst König represents prefixes by labelling edges with process graphs, we instead exploit the hierarchical nature (viz. nesting) of bigraphs. The work is extended in [11] to also account for simple sorting in the style of Milner.

Technical contributions. The main technical contributions of this paper are as follows.

- ▶ We lift the theory of [8] to a sorted setting (§3), and generalise the definition of reflect pushouts (Definition 4).
- ▶ We introduce a sorting (Definition 6) which represents the capability types, and prove that it satisfies the requirements for sortings (Theorem 4).
- ▶ We derive and characterise labels (Lemma 3) using the bigraphical theory of *relative pushouts*, and prove that we can limit our attention to labels that do not introduce redundant subsorting (Corollary 3).

Due to space limitation we have placed proofs and additional formal work in the full version of this paper [2].

Structure of the paper. In §1 we recall the notion of binding bigraph and their behavioural theory. The reader is referred to the appendix – and ultimately to the relevant literature – for a formal introduction. We present link-sortings in §3. In §4 we review the polyadic π -calculus with capability types and subtyping, and in §5 present it as a sorted bigraphical reactive system. We conclude and propose further work in §6.

1. Binding bigraphs

We briefly and informally review the binding bigraphs of Milner and Jensen [8, §11] which we generalise to include signatures and sorts for edges. A more formal introduction including the definition

of pure bigraphs and their constituting place and link graphs is given in Appendix A.

We define bigraphical reactive systems with respect to a binding signature and to an edge signature. A *binding signature* \mathcal{K} consists of a set of *controls* $K \in \mathcal{K}$, each equipped with a pair of finite ordinals: a *binding* arity h and a *free* arity k , in symbols $K: h \rightarrow k$. Controls represent the ‘node-types’ that may occur in a bigraph, and the arities of K indicate respectively the number of binding and of free *ports* of a bigraph’s K -nodes. We set $ar(K) = h + k$. Signatures also dictate which controls are *atomic* (i.e. may contain no nested bigraph), and declare each non-atomic control to be either *active* (i.e. allowing internal reactions to occur) or *passive*. As every node in a bigraph is associated with a control, a binding signature completely prescribes the number of ports of each node, whether they are binding or not, whether the node can contain other nodes, and whether reaction can occur within the node. For instance, with reference to the binding bigraph of Fig. 1, nodes v_4 and v_0 have a arity-one control, whilst the control associated to v_2 must have arity $1 \rightarrow 1$ (the binding port being represented by the small circle on the node).

An *edge signature* \mathcal{E} is simply a set of *edge controls*. At the moment we associate no further information to edge controls; they should therefore be thought as elementary edge-types that we will use in §3 to sort edges and express conditions on them.

A *binding interface* is a triple $\langle m, \vec{X}, X \rangle$, where m – the *width* – is a finite ordinal representing a list of sites or regions, X is a finite set of names, and \vec{X} – the *locality* map – is a m -tuple of pairwise disjoint subsets of X which declares some of the names in X as local to specific places in m . If $x \notin \vec{X}$ then x is said to be *global*. Binding interfaces describe bigraphs’ external connectivity in terms of number of regions and local/global names. Each bigraph G has both an *inner face* and an *outer face*; the former lists requirements on bigraphs to be inserted in G , the latter has the dual role of describing G as a candidate for insertion in a larger bigraphical context.

A *pure* bigraph is a bigraph with nor local names, i.e. whose faces have trivial locality maps, nor binding ports. It consists of the superimposition of two graph structures sharing a set nodes: the place graph and the link graph (cf. Appendix A). The *place graph* is determined by a *parent* map, which describes the nesting of nodes and gives rise to an ordered forest of trees. The number of trees is determined by the width of the outer face; the number of sites by the width of the inner face. The *link graph* is described by a *link* map which connects inner names and nodes’ ports to outer names and edges. Both nodes and edges are associated to their respective controls \mathcal{K} and \mathcal{E} by a function *ctrl*.

Bigraphs with compatible interfaces and disjoint node and edge sets can be composed. The composition $H \circ G: I \rightarrow I'$ of $G: I \rightarrow I'$ and $H: I' \rightarrow I''$ is obtained by replacing the H ’s i th site with G ’s i th region and by coalescing links as prescribed by the correspondence of H ’s inner and G ’s outer names. The tensor product $G \otimes H$ is defined for bigraphs with disjoint node, edge and name sets, giving rise to the bigraph that can be drawn simply by putting G and H side by side. As tensor and composition are partial operations, this yields a so-called *precategory* (more precisely, a *s-category* [8]). The following definition lifts these concepts to binding bigraphs.

DEFINITION 1 (BINDING BIGRAPHS). The precategory $\mathcal{B}BG(\mathcal{K}, \mathcal{E})$ of concrete binding bigraphs over signatures \mathcal{K} and \mathcal{E} has binding interfaces as objects and binding bigraphs as arrows. A concrete *binding bigraph* $G: \langle m, \vec{X}, X \rangle \rightarrow \langle n, \vec{Y}, Y \rangle$ consists of an underlying pure bigraph $G^\mu: \langle m, (), X \rangle \rightarrow \langle n, (), Y \rangle$ with extra structure as follows. Define the *binders* of G to be the binding ports of its nodes, as defined by *ctrl* and \mathcal{K} , together with the local names of its outer face J (those in \vec{Y}). Then G must satisfy the following:

SCOPE RULE: If p is a binder located at a node or, in the case of local names, at a root w , then every p' sharing the same link as p must be located at a place w' (a site, root or node) such that $w' <_{G^u} w$, i.e. w' is a descendent of w in the nesting determined by (the place graph underlying) G .

If G and H are concrete binding bigraphs with compatible interfaces and disjoint support (i.e., node and edge sets), then their composition \circ is defined in terms of composition of the underlying pure bigraphs. Similarly, for the tensor product \otimes when, in addition, inner and outer name sets are disjoint.

Bigraph terminology. There is quite a lot of terminology to digest about bigraphs. Here we introduce selectively the main notions we need in the paper.

A *point* is either a port or an inner name. A *link* is either an edge or outer name. A link is *open* if it is a name, *closed* if it is an edge. It is *bound* if it is a local name or is connected by the link map to a binder; otherwise we say it is *free*. These terms are extended to the points in the link.

An interface is *prime* if it has width 1. A binding bigraph $G: I \rightarrow J$ is *prime*, if all names in I are local and J is prime. A bigraph is *ground* if it has unit inner face $\epsilon = \langle 0, (), \emptyset \rangle$; these bigraphs represent agents, whereas bigraphs with holes represents contexts. A binding bigraph is *discrete* if every free link is a name and has exactly one point. We call an edge *idle* if no point is connected to it. A bigraph is *lean* if it has no idle edges. The bigraph in Fig. 2 is ground, lean and discrete, as it has no idle edges and its free ports are in a 1-1 correspondence with outer names; its only edge in the bigraph is bound by the binding port of v_2 . It is an important result of [8] that bigraphs can be expressed uniquely up-to-isomorphism as a composition of a discrete bigraph and a wiring – i.e. a link graph with no nodes – whose purpose is to reconnect the free links so as to obtain the original edges and name sharing.

Ideally, bigraphs that only differ from the identity of edges and nodes should be equated, as these are conceptually irrelevant. We say that G and H are *lean-support* equivalent, in symbols $G \approx H$, if they are isomorphic after discarding all idle edges. An *abstract binding bigraph* is a lean-support equivalence class of concrete binding bigraphs. For any signatures \mathcal{K} and \mathcal{E} we have a category $\text{BBG}(\mathcal{K}, \mathcal{E})$ and the quotient functor

$$\llbracket \cdot \rrbracket : \text{BBG}(\mathcal{K}, \mathcal{E}) \rightarrow \text{BBG}(\mathcal{K}, \mathcal{E}) .$$

Observe that abstract bigraphs form a category. The reason that prevents us from working in $\text{BBG}(\mathcal{K}, \mathcal{E})$ is that the latter does not possess relative pushouts (RPOs; cf. Appendix B) which is the fundamental notion on which the dynamic theory of bigraphs rests. The approach developed by Milner, Leifer, and Jensen that we follow in this paper is to work in the precategory of concrete bigraphs – where RPOs exist – and then transfer the results to abstract bigraphs along the quotient functor $\llbracket \cdot \rrbracket$. A parallel approach has been developed in [18, 19, 9] based on the use of *bicategories*.

Bigraph term language. Bigraphs admit an elegant and fruitful representation via a simple term language; this is an alternative to their graphical representation, which in the future may be exploited to develop executable specifications. For each non-atomic control $K: h \rightarrow k$ and every choice of $h + k$ distinct names \vec{y} and \vec{x} , the term language contains a *ion* $K_{\vec{y}\vec{x}}$ with inner local names \vec{y} , free outer names \vec{x} and one site (hole). Similarly, for an atomic control K we have ground atoms like $K_{\vec{x}}$. (Note that terms are subject to α -conversion of inner local names.) Wiring can be obtained from elementary terms like $/x$, x/X , and $\ulcorner X \urcorner$, representing respectively the closure of a name x , the substitution of x for a set of names X , and the concretion or ‘globalisation’ of local names X .

Besides composition \circ and tensor \otimes , terms can be combined by *parallel product* $g \parallel h$ and *prime parallel product* $h \mid g$ which,

differently from \otimes , are defined even if h and g share outer names. precisely, \parallel works like \otimes but links the common outer names, while \mid merges the multiple regions resulting from \parallel to a single one. The denotations of these combinators as operations on concrete bigraphs are described in Appendix A.

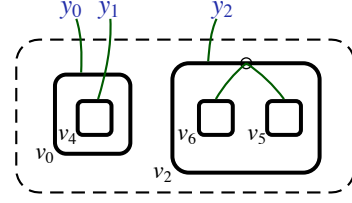


Figure 2. A discrete bigraph

The bigraph in Fig. 2 above can be represented in the term language as below, where for the sake of example we assume that node v_i has control K^i .

$$(K_{y_0}^0 \otimes \text{id}_{y_1}) \circ K_{y_1}^4 \mid K_{y_2(z)}^2 \circ (K_z^6 \mid K_z^5)$$

Or, omitting \circ and id s as customary: $K_{y_0}^0 (K_{y_1}^4) \mid K_{y_2(z)}^2 (K_z^6 \mid K_z^5)$.

Reactions and transitions. The dynamics of bigraphical reactive systems is defined in terms of a reaction relation on $\text{BBG}(\mathcal{K}, \mathcal{E})$ generated from a set of reaction rules \mathcal{R} . Such rules are usually parametric of a very general nature, and may discard and also duplicate their parameters. This must be handled with care, because of name sharing between parameters. We describe next how to derive ground reaction rules from parametric ones by instantiating with ground parameters. As in [8], it suffices to limit ourselves to *discrete* parameters and *affine* instantiations, i.e. instantiations that do not duplicate parameters.

A *ground (reaction) rule* is a pair (r, r') , where r and r' are ground bigraphs with the same outer face. Given a set of ground rules, the reaction relation \rightarrow over agents is the least relation such that $D \circ r \rightarrow D \circ r'$ for each ground rule (r, r') and each active context D – i.e., contexts whose hole is nested only inside active controls.

A *parametric (reaction) rule* has a *redex* R and a *reactum* R' , and takes the following form

$$(R: I \rightarrow J, R': I' \rightarrow J, \varrho) ,$$

where the inner faces I and I' are local with widths m and m' , and ϱ is an instantiation function generated by an injective, not necessarily surjective function $\bar{\varrho}: m' \rightarrow m$. The latter is meant to map the m parameters in the redex to the m' parameters in the reactum according to the correspondence prescribed by $\bar{\varrho}$. Precisely, any discrete $d: \epsilon \rightarrow X \otimes I$ is a tensor of discrete primes $d = d_0 \otimes \dots \otimes d_{m-1}$. Essentially, $\varrho(d)$ is their combination under \parallel (i.e., with sharing of outer names) in $\bar{\varrho}$ -order, $\varrho(d) = X \parallel d_{\bar{\varrho}(0)} \parallel \dots \parallel d_{\bar{\varrho}(m'-1)}$, worked out up to lean-support isomorphism in order to guarantee that the product is well defined. Since some d_i may be dropped, the resulting outer face may not have all the required names, which is the reason to add the factor $X: \epsilon \rightarrow \langle 0, (), X \rangle$. Summing up, instantiating the parametric rule with d generates the ground reaction rule

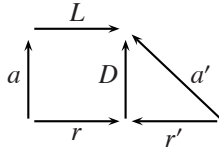
$$((\text{id}_X \otimes R) \circ d, (\text{id}_X \otimes R') \circ \varrho(d)) .$$

2. Basic bigraphical reactive systems

In [8, §14], *basic bigraphical reactive systems* are introduced as a specially constrained class of BRSs which allow for a simple characterisation of the labels of the standard transition systems generated from them via the relative pushout construction. Furthermore, for basic BRS one can focus attention solely on certain simple transitions without affecting the resulting labelled bisimilarity.

The conditions which define a basic BRS enforce important simplifying properties of redexes. Informally, these include ideas such as flatness (no nesting of nodes), guardedness (no inner name is open, no site has a root as parent), simpleness (no inner names are peer, no sites are siblings), and definiteness (no redex involve only a subset of the controls involved in another). Together, they enforce nice categorical properties such as that the parametric redexes are both epi and mono. We refer the reader to the full paper for a formal definition of these requirements, and here we content ourselves with saying that basic BRS yield a tractable transition system whose bisimilarity is a congruence. And of course, that fortunately they are expressive enough to represent the polyadic π -calculus of this paper.

For an agent a and a ground reaction rule (r, r') we aim at deriving a *standard* transition $a \xrightarrow{L} a'$, as expressed in the diagram below where L and D are an idem pushout of a and r (informally, think pushout; cf. Appendix B).



Note that D must be an active context. We call the transition system of a BRS consisting of all standard transitions the *standard transition system*, denoted ST , and write \sim_{ST} for the associated bisimilarity.

LEMMA 1. *In any BRS equipped with the standard transition system, bisimilarity of (concrete) ground bigraphs is a congruence.*

We say that one transition system \mathcal{M} is a *sub-lts* of \mathcal{L} whenever the interfaces and transitions of \mathcal{M} are included in the interfaces and transitions of \mathcal{L} .

DEFINITION 2 (RELATIVE BISIMILARITY, ADEQUACY). Given a transition system \mathcal{L} and a sub-lts \mathcal{M} . We define a *relative bisimulation* for \mathcal{M} as a symmetric relation \mathcal{S} such that whenever $a \mathcal{S} b$, then for every transition $a \xrightarrow{L} a'$ in \mathcal{M} where $L \circ b$ is defined, there exists b' such that $b \xrightarrow{L} b'$ in \mathcal{L} and $a' \mathcal{S} b'$. We define *relative bisimilarity* for \mathcal{M} , written $\sim_{\mathcal{L}}^{\mathcal{M}}$, as the largest relative bisimulation for \mathcal{M} (on \mathcal{L}). We say that \mathcal{M} is *adequate* for \mathcal{L} when $\sim_{\mathcal{L}}^{\mathcal{M}}$ coincides with $\sim_{\mathcal{L}}$ on the agents in \mathcal{M} .

We will focus our attention on *engaged transitions* of prime, ground bigraphs with free outer face. These are transitions where the agent a shares at least one node with the parametric redex R underlying the ground r , and the reason for the restriction is to match our intuition that in order for a transition to be significant, a must actually contribute to it. We write FPE for the sub-lts of ST consisting of engaged transitions and \sim_{FPE} for its bisimilarity; we write $\sim_{\text{ST}}^{\text{FPE}}$ for the relative bisimilarity for FPE on ST .

THEOREM 1 (cf. [8]). *In any basic BRSs FPE is adequate for ST and \sim_{FPE} coincides with \sim_{ST} and is therefore a congruence.*

Transferring engaged transitions. In order to transfer \sim_{FPE} to abstract bigraphs via a quotient functor, we need to enforce more structure on the concrete bigraphs where we derive the behavioural equivalence. As in [8], we find it convenient to work in $\mathcal{B}\text{BG}_h(\mathcal{K}, \mathcal{E})$, the sub-precateory of $\mathcal{B}\text{BG}(\mathcal{K}, \mathcal{E})$ consisting *hard bigraphs*. These are bigraphs whose place graphs are epi, and can be characterised as those where all non-atomic nodes and roots contain at least one node.

We define *place equivalence* as the least equivalence which equate bigraphs that only differ on nodes with the special atomic control Δ with zero arity. *Soft lean-support equivalence*, written \approx_{Δ} ,

is then taken to be the smallest equivalence including lean-support equivalence and place equivalence. It is easy to prove that we have a \approx_{Δ} -quotient functor

$$\llbracket \cdot \rrbracket_{\Delta} : \mathcal{B}\text{BG}_h(\mathcal{K}^{\Delta}, \mathcal{E}) \rightarrow \text{BBG}(\mathcal{K}, \mathcal{E}) ,$$

where \mathcal{K}^{Δ} stands for the signature \mathcal{K} extended with the fresh control Δ . Note that we can make any bigraph in $\mathcal{B}\text{BG}(\mathcal{K}, \mathcal{E})$ into a bigraph in $\mathcal{B}\text{BG}_h(\mathcal{K}^{\Delta}, \mathcal{E})$ by placing Δ -nodes in all roots and non-atomic nodes which break the hard requirement.

Thanks to the properties of basic BRS we can now transfer the bisimilarity \sim_{FPE} for free prime engaged transitions from the hard concrete setting to the (soft) abstract setting while the induced bisimilarity remains a congruence.

THEOREM 2 (cf. [8]). *Let $\mathcal{B}\text{BG}_h(\mathcal{K}^{\Delta}, \mathcal{E})$ be a hard concrete basic BRS whose all redexes are Δ -free. Then*

1. $a \sim_{\text{FPE}} b$ in $\mathcal{B}\text{BG}_h(\mathcal{K}^{\Delta}, \mathcal{E})$ iff $\llbracket a \rrbracket_{\Delta} \sim_{\text{FPE}} \llbracket b \rrbracket_{\Delta}$ in $\text{BBG}(\mathcal{K}, \mathcal{E})$.
2. \sim_{FPE} is a congruence in $\text{BBG}(\mathcal{K}, \mathcal{E})$.

3. Sorted binding bigraphs

We are now ready to define link-sorted binding bigraphs generalising the idea of [13] to also include sorts to edge signatures. Using the same notation as *loc. cit.*, we let Θ denote a non-empty set of *sorts* and use θ to range over Θ .

We enrich the edge signature \mathcal{E} by ascribing a sort in Θ to each edge control. We say that \mathcal{E} is Θ -sorted. A binding bigraph on \mathcal{K} and \mathcal{E} is Θ -sorted if both its interfaces are enriched by ascribing a sort to each name. We differ from previous work by say that a port p has sort S , when p points to a link with sort S , i.e. ports inherit their sorting from the link they are connected to. We denote by $\mathcal{S}\text{BBG}_h(\Theta, \mathcal{K}, \mathcal{E})$ the monoidal precategory of sorted hard binding bigraphs whose identities, composition, and tensor product are defined in terms of the underlying binding bigraphs.

DEFINITION 3 (SORTING DISCIPLINE). A *sorting* (discipline) is a tuple $\Sigma = (\Theta, \mathcal{K}, \mathcal{E}, \Phi)$, where \mathcal{E} is Θ -sorted and Φ is a condition on Θ -sorted binding bigraphs over \mathcal{K} and \mathcal{E} . The condition Φ must be satisfied by the identities and preserved by both composition and tensor product.

We say that a binding bigraph in $\mathcal{S}\text{BBG}_h(\Theta, \mathcal{K}, \mathcal{E})$ is Σ -sorted if it satisfies Φ . Often, when Σ is understood from the context, we omit Σ and just call Σ -sorted bigraph *well-sorted*. The Σ -sorted binding bigraphs form a monoidal sub-precateory of $\mathcal{S}\text{BBG}_h(\Theta, \mathcal{K}, \mathcal{E})$ denoted by $\mathcal{S}\text{BBG}_h(\Sigma)$. If \mathcal{R} is a set of Σ -sorted reaction rules, $\mathcal{S}\text{BBG}_h(\Sigma, \mathcal{R})$ is a Σ -sorted BRS.

Together with the sorting we of course have a forgetful functor which we call a *sorting* functor $\Sigma : \mathcal{S}\text{BBG}_h(\Sigma) \rightarrow \mathcal{B}\text{BG}_h(\mathcal{K}, \mathcal{E})$. Like the forgetful functor from binding bigraphs to pure bigraphs the forgetful functor induced by the sorting is faithful.

DEFINITION 4 (CREATING RPOS, WEAKLY REFLECTING PUSHOUT). Let \mathcal{F} be a functor on a precategory \mathbf{A} , and let i range over $\{0, 1\}$. Then \mathcal{F} *creates RPOs* if, whenever D_i bounds A_i in \mathbf{A} , any RPO for $\mathcal{F}(A_i)$ relative to the bound $\mathcal{F}(D_i)$ has a unique \mathcal{F} -preimage that is an RPO for A_i relative to D_i .

\mathcal{F} *weakly reflects pushouts* if, whenever D_i is an IPO for A_i in \mathbf{A} and $\mathcal{F}(D_i)$ is a pushout for $\mathcal{F}(A_i)$, then D_i is a pushout for A_i .

Here we weaken the definition used in [13] by only requiring that pushouts are reflected for IPOs in the sorted precategory, rather than for all possible bounds. In particular, while our sorting does not reflect pushouts for all possible bounds, it does so for IPOs. The reason we need this is to ensure that the codomain of the cospan D_i is the 'least' possible (e.g. in terms of idle links) in order for

D_i to be a pushout for A_i . This slight change of definition allows for a greater range of sortings while still enjoying the benefits of a tractable transition system based on engaged transitions.

Sorted binding bigraphs support a notion of basic BRS analogous to the one described for (unsorted) binding bigraphs in the previous section. We can prove that the sorting functor reflects and preserves this property, and therefore retrace the development in §2. The following theorem matches very closely the one in [13], and ensures the existence of RPOs and guarantees that FPE coincides with \sim_{ST} on free prime interfaces when the sorting functor satisfies the conditions of Definition 4.

THEOREM 3 (USEFUL SORTINGS). In $\mathcal{S}\text{BBG}_h(\Sigma, \mathcal{R})$

- If Σ creates RPOs, then bisimilarity for the standard transition system, \sim_{ST} , is a congruence.
- If in addition Σ weakly reflects pushouts and \mathcal{R} is basic, then $\sim_{\text{ST}}^{\text{FPE}}$ is adequate for \sim_{ST} .

As in the previous section we let

$$\llbracket \cdot \rrbracket_{\Delta} : \mathcal{S}\text{BBG}_h(\mathcal{K}^{\Delta}, \mathcal{E}) \rightarrow \text{SBBG}(\mathcal{K}, \mathcal{E})$$

be the quotient functor by \approx_{Δ} , and we can again transfer the congruential bisimilarity \sim_{FPE} from the hard concrete setting to the soft abstract setting.

COROLLARY 1. Let $\mathcal{S}\text{BBG}_h(\mathcal{K}^{\Delta}, \mathcal{E})$ be a hard sorted concrete BRS that is basic and whose all redexes are Δ -free. Then

1. $a \sim_{\text{FPE}} b$ in $\mathcal{S}\text{BBG}_h(\mathcal{K}^{\Delta}, \mathcal{E})$ iff $\llbracket a \rrbracket_{\Delta} \sim_{\text{FPE}} \llbracket b \rrbracket_{\Delta}$ in $\text{SBBG}(\mathcal{K}, \mathcal{E})$.
2. \sim_{FPE} is a congruence in $\text{SBBG}(\mathcal{K}, \mathcal{E})$.

We have now taken the theory of sorted binding bigraphs as far as required to provide a treatment of the polyadic π -calculus with a tractable labelled transition system and a congruential bisimilarity.

Sorting for the polyadic π -calculus

In this subsection we present a sorting for binding bigraphs, *sub-sorting*, where we require that the sort of a point connected to a link is a subsort of the link's sort, and that lower bounds can be expressed on sets of sorts. Most of the constructions in this section require the existence of bounded meets between pair of sorts.

DEFINITION 5 (BOUNDED MEETS). Let \mathcal{S} be a set of sorts and \leq a preorder on \mathcal{S} . The *meet* of a pair of sorts T and T' , written $T \sqcap T'$, is a sort S such that $S \leq T$, $S \leq T'$, and for all other sorts S' with the same property, we have $S' \leq S$. We say that the preorder has *bounded meets* if whenever $U \leq T$ and $U \leq T'$, then there exists a meet of T and T' .

In the definition below, function *pack* is responsible for combining a list of sorts and a type constructor from a set Q to return the sort resulting from their combination. A constructor $q \in Q$ can be *covariant*, *contravariant* or *invariant* on each of its arguments. We use this information to derive a preorder relation \leq on $\mathcal{S}^* \times Q$; namely, for $S \leq T$ and a q covariant on i we set $(\dots, S, \dots, q) \leq (\dots, T, \dots, q)$, where S is the i th element in the tuple; dually, $(\dots, T, \dots, q) \leq (\dots, S, \dots, q)$ if q is contravariant on i . We also assume a non-commutative partial multiplication \times on Q . This induces a partial multiplication $(\vec{T}, q) \times (\vec{S}, q')$ defined as $(\vec{T}, \vec{S}, q \times q')$ exactly when $q \times q'$ is, and $\vec{S} \leq \vec{T}$ componentwise. We then augment our preorder by stipulating that $(\vec{T}, \vec{S}, q \times q')$ precedes both (\vec{T}, q) and (\vec{S}, q') . Function *pack* will be required to be monotonic with respect to such a preorder structure.

DEFINITION 6 (SUBSORTING). A *subsorting* $\Sigma = (\Theta, \mathcal{K}, \mathcal{E}, \Phi)$ is a set \mathcal{S} with a preorder \leq with bounded meets, together with a partial

injective monotonic function *pack*: $\mathcal{S}^* \times Q \rightarrow \mathcal{S}$, which takes a tuple of sorts and a type constructor q and returns a sort. Every control $K \in \mathcal{K}$ with $\text{ar}(K) = n > 0$ is associated with a type constructor $q_K \in Q$ and a partition of n into two sets, C_K and \mathcal{V}_K , of respectively *communication* and *value* ports. If q is covariant on i and K 's i th port is a value port, then it must be a *binding* port. The edge signature \mathcal{E} delivers an arbitrary assignment of sorts to edge controls, and the condition Φ is as follows:

- For each inner name x : S , if T is the sort of its link, then $T \leq S$.
- For each K -node v and each $c \in C_K$, let S be the sort of v 's corresponding port; we require that $S \leq \text{pack}(S_1, \dots, S_{|\mathcal{V}_K|}, q_K)$, where S_i is the sort of v 's i th port in \mathcal{V}_K .

Intuitively the first requirement ensures that we can always use subsumption on sorts, while the second requirement allows us to express lower bounds on the combination of a tuple of sorts under the appropriate type constructor. The condition about binding ports guarantees well-sortedness when subsorting under a covariant constructor. It is straightforward to prove that subsorting is satisfied by identities and preserved by composition and tensor product.

In the following we will use \mathcal{U}_s to denote the subsorting functor which maps a well-sorted bigraph to the underlying binding bigraph. Functor \mathcal{U}_s does not reflect pushout in the sense of [13]. It is however possible to prove that \mathcal{U}_s weakly reflects pushouts, since the outer interface of an IPO is the smallest possible. We can show the following results for subsorting.

THEOREM 4. The subsorting functor \mathcal{U}_s creates RPOs and weakly reflects pushouts.

Most of the concepts of §1 can be transferred to subsorted bigraphs with only minor changes. Notably, for parallel product \parallel and prime product $|$ we require that shared names have a common subsort and in the resulting outer face we associate to them the meet of their sorts. Instantiation and reaction rules can then be transferred to the sorted setting *mutatis mutandis*.

4. Polyadic pi-calculus

In this section we briefly introduce the polyadic π -calculus of [14] with a variant of Pierce and Sangiorgi's type system [17], where we have capability types and a subsorting relation with bounded meets as in [5, 4]. We assume an infinite set of names \mathcal{N} and let m, n range over \mathcal{N} . We let P and Q range over process expressions, and S, T , and U over sorts. The set \mathcal{P}_{π} of *process expressions* is defined by the following grammar

$$P ::= \mathbf{0} \mid P \mid P' \mid (vn : S)P \mid \bar{n}(m_1, \dots, m_i).P \mid n(m_1 : S_1, \dots, m_i : S_i).P .$$

We use $\mathcal{P}_{\pi/\alpha}$ to denote the set of α -equivalence classes of process expressions, as we consider processes up to α -equivalence. We write $(v\tilde{n} : \vec{S})P$ for the restriction in process P of a set of names \tilde{n} with associated sorts \vec{S} .

For brevity we will not define structural congruence in this paper. We remark however that we include the following rule, for π a generic prefix: $\pi.(vn : S)P \equiv_{\pi} (vn : S)\pi.P$, if $n \notin \text{fn}(\pi)$. An *evaluation context* is a context whose hole is not under a prefix. We write $P\{m'_1, \dots, m'_i / m_1, \dots, m_i\}$ for the simultaneous capture-avoiding substitution of \tilde{m}'_j for \tilde{m}_j in P . The *reaction relation* \rightarrow_{π} is the least binary relation over $\mathcal{P}_{\pi/\alpha}$ satisfying the following rule and closed under evaluation contexts and structural congruence

$$\frac{}{n(m_1 : S_1, \dots, m_i : S_i).P \mid \bar{n}(m'_1, \dots, m'_i).Q \rightarrow_{\pi} P\sigma \mid Q}$$

where $\sigma = \{m'_1, \dots, m'_i / m_1, \dots, m_i\}$.

$$\begin{array}{c}
\frac{T_i \leq T'_i, \quad i = 1, \dots, n}{(T_1, \dots, T_n)^f \leq (T'_1, \dots, T'_n)^f} \quad \frac{T_i \leq T'_i, \quad i = 1, \dots, n}{(T_1, \dots, T_n)^w \leq (T'_1, \dots, T'_n)^w} \quad \frac{T_i \leq T'_i \text{ and } S_i \leq S'_i, \quad i = 1, \dots, n}{(T_1, \dots, T_n; S'_1, \dots, S'_n)^b \leq (T'_1, \dots, T'_n; S_1, \dots, S_n)^b} \\
\frac{T_i \leq T'_i, \quad i = 1, \dots, n}{(T_1, \dots, T_n; S_1, \dots, S_n)^b \leq (T'_1, \dots, T'_n)^f} \quad \frac{T_i \leq T'_i, \quad i = 1, \dots, n}{(S_1, \dots, S_n; T'_1, \dots, T'_n)^b \leq (T_1, \dots, T_n)^w}
\end{array}$$

Figure 3. The subsort relation

$$\begin{array}{c}
\frac{}{\Gamma \vdash \mathbf{0} : \circ} \quad \frac{\Gamma \vdash P : \circ \quad \Gamma \vdash Q : \circ}{\Gamma \vdash P \mid Q : \circ} \quad \frac{\Gamma, n : S \vdash P : \circ}{\Gamma \vdash (\nu n : S)P : \circ} \\
\frac{\Gamma(n) \leq (S_1, \dots, S_n)^f \quad \Gamma, m_1 : S_1, \dots, m_n : S_n \vdash P : \circ}{\Gamma \vdash n(m_1 : S_1, \dots, m_n : S_n).P : \circ} \quad \frac{\Gamma(n) \leq (\Gamma(m_1), \dots, \Gamma(m_n))^w \quad \Gamma \vdash P : \circ}{\Gamma \vdash \bar{n}(m_1, \dots, m_n).P : \circ}
\end{array}$$

Figure 4. The typing judgement

Now we introduce a sorting mechanism inspired by [5, 4]. We use I to range over *input/output tags* defined as $I ::= r \mid w \mid b$, which represent respectively the receive capability, the send capability, and the capability to both send and receive. The set of sorts \mathcal{S} of our type system is defined by the following rules.

$$\begin{array}{c}
\frac{}{\circ^f :: \text{Type}} \quad \frac{T_1 \dots T_n :: \text{Type}}{(T_1, \dots, T_n)^f :: \text{Type}} \quad \frac{T_1 \dots T_n :: \text{Type}}{(T_1, \dots, T_n)^w :: \text{Type}} \\
\frac{T_1 \dots T_n :: \text{Type} \quad S_1 \dots S_n :: \text{Type} \quad S_i \leq T_i}{(T_1, \dots, T_n; S_1, \dots, S_n)^b :: \text{Type}}
\end{array}$$

The most interesting rule is the one for the **b**-tag. In a sort $(T_1, \dots, T_n; S_1, \dots, S_n)^b$ the first set of components, T_1, \dots, T_n , captures the sort's input capability, whilst the second set, S_1, \dots, S_n , captures the output capability. The side-condition on rule $S_i \leq T_i$, which uses the subsort relation (cf. below) ensures that the communicated output on a channel belongs to a subsort of the expected input sort. We will omit the annotation $S :: \text{Type}$ on sorts.

A *sorting* Γ is a finite partial mapping from names to sorts; writing $\Gamma, a : S$ we tacitly assume that $a \notin \Gamma$. Clearly, the order of bindings in Γ is immaterial. We introduce a subsort relation, \leq , on sorts as the least preorder satisfying the rules of Fig. 3. The rules express that the **r**-tag (resp. **w**-tag) is a covariant (resp. contravariant) constructor. On the other hand, the **b**-tag operator is covariant on the first set of components and contravariant on the second. The typing judgement $\Gamma \vdash P : \circ$ in Fig. 4 says that the process P is well-sorted under the assumption Γ , which must contain all the free names of P . The rules are self-explanatory, but note that the rules for input and output checks that the sort of the name n is a subsort of the expected sort. This type system enjoys the following standard result.

LEMMA 2 (SUBJECT REDUCTION). *If $\Gamma \vdash P : \circ$, then $\Gamma \vdash P' : \circ$ for each $P \rightarrow_{\pi} P'$.*

5. Bigraphical representation

We now proceed to represent the polyadic π -calculus and its subsorting as a bigraphical reactive system. Our encoding slightly extends Jensen's presentation of the synchronous monadic π -calculus in [6], which in turn is based upon Jensen and Milner's encoding of the asynchronous π -calculus in [7, 8]. Firstly and foremost, we straightforwardly extend the presentation from the monadic π -calculus to the polyadic π -calculus. We then consider the *sorted* polyadic π -calculus with capability types and subtyping.

In the following we will write Γ_b for the empty ground bigraph with the outer face $\langle 1, () \rangle$, $\text{dom}(\Gamma)$, Γ , interpreting Γ as a function from names to sorts; $\text{sub}_X : \langle 1, () \rangle, X, \text{sort} \rangle \rightarrow \langle 1, () \rangle, X, \text{sort}' \rangle$ will denote the substitution that acts as the identity on the set of names X , whilst possibly changing the sort of some of them to a subsort, i.e. $\text{sort}'(x) \leq \text{sort}(x)$, for all names $x \in X$. Finally, we write $\langle \text{sort} \rangle$ as a shorthand for the free prime interface $\langle 1, () \rangle, \text{dom}(\text{sort}), \text{sort} \rangle$.

DEFINITION 7 (SORTING). Sorting is an instance of subsorting in Definition 6. The set of sorts \mathcal{S} is the one defined in the previous section, while Q is the set of tags $\{b, r, w\}$ with $r \times w = b$ and

$$\text{pack}(S_0, \dots, S_n, q) = (S_0, \dots, S_n)^q,$$

defined when the target type exists. The signature will have countably many **send** and **get** controls. All **send** controls are associated with **w**, and their value ports are contravariant; **get** controls with **r**, and their value ports are covariant (and binding).

DEFINITION 8 (SORTED BRS). The sorted BRS

$$\text{SBBG}_{\pi_{\leq}} = \text{SBBG}_{\pi_{\leq}}(\Sigma_{\pi_{\leq}}, \mathcal{R}_{\pi_{\leq}})$$

has signatures $\mathcal{K}_{\pi_{\leq}}$ and $\mathcal{E}_{\pi_{\leq}}$ defined as follows. The edge signature $\mathcal{E}_{\pi_{\leq}}$ provides a set of controls in one-to-one correspondence with the sorts in \mathcal{S} . The node signature $\mathcal{K}_{\pi_{\leq}}$ consists of two infinite families of non-atomic passive controls, for i a finite ordinal,

$$\text{send} : 0 \rightarrow (i + 1) \quad \text{get} : i \rightarrow 1.$$

The rule set $\mathcal{R}_{\pi_{\leq}}$ consists of a family of Σ -sorted reaction rules depicted in Fig. 6. The outer names y_1, \dots, y_n have sorts T_1, \dots, T_n , the local names z_1, \dots, z_n and the edges they are connected to have sort $(U_1, \dots, U_n; T_1, \dots, T_n)^b$.

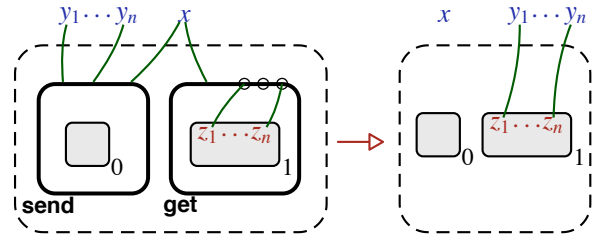


Figure 6. Reaction rules for polyadic π -calculus

The **send** and **get** controls with index i and types \vec{U} and \vec{T} represent respectively the output and input prefix exchanging a tuple of

$$\begin{aligned}
\llbracket \Gamma \vdash \mathbf{0} : \circ \rrbracket &= \Gamma_b \\
\llbracket \Gamma \vdash P \mid Q : \circ \rrbracket &= \llbracket \Gamma \vdash P : \circ \rrbracket \mid \llbracket \Gamma \vdash Q : \circ \rrbracket \\
\llbracket \Gamma \vdash (vn : S)P : \circ \rrbracket &= (/n : S) \llbracket \Gamma, n : S \vdash P : \circ \rrbracket \\
\llbracket \Gamma \vdash \bar{n}(m_1, \dots, m_i).P : \circ \rrbracket &= \mathbf{send}_{n, m_1, \dots, m_i}(\llbracket \Gamma \vdash P : \circ \rrbracket) \\
\llbracket \Gamma \vdash n(m_1 : S_1, \dots, m_i : S_i).P : \circ \rrbracket &= \mathbf{get}_{n(m_1, \dots, m_i)}(m_1, \dots, m_i)(\llbracket \Gamma, m_1 : S_1, \dots, m_i : S_i \vdash P : \circ \rrbracket)
\end{aligned}$$

Figure 5. Translation of a well-typed process $\Gamma \vdash P : \circ$

length i of appropriate type. Observe that in order for the sort of x to be well-formed, $T_i \leq U_i$ must hold. Note also that we require the input and the output sorts to match precisely the sort of x , the name carrying the message. This ‘minimal’ description induce a large set of reaction rules, since the context of the reaction can use subsumption to obtain reactions where the sorts do not match precisely. We can therefore apply the reaction rule to an agent in any well-sorted setting, as long as a **send** node is linked to a **get** node.

Even though the rules are specified using a minimal description they are not as good as one would desire, as they give rise to too many transitions, as e.g. transitions that introduce redundant subsorting. We conjecture that it is not possible to identify a complete set of reaction rules which does not give rise to such transitions. An obvious attempt would be to assume the existence of minimal and maximal sorts and give the outer names y_i the maximal sort, which can then subsort correctly in the context. However this will only work if the sorts of z_i are also given the maximal sort, as otherwise the rule will not be well-typed. So even with minimal and maximal sorts we could not describe the reaction rules without indexing over the types. Nevertheless, we will prove that we can limit our attention to the transitions which introduce no redundant subsorting, and still obtain the same congruence.

In Fig. 5 we define a translation $\llbracket - \rrbracket$ of a well-typed process $\Gamma \vdash P : \circ$ to bigraphs inductively on the typing derivation of P . The target set is the homset $\epsilon \rightarrow \langle \Gamma \rangle$ of $\mathbb{SBBG}_{\pi \leq}$. We map the inactive process to the empty bigraph with the correct outer interface; parallel composition is translated into a prime product; restriction is mapped to a name closure of the appropriate sort; and prefixes are translated into a node of the respective controls.

PROPOSITION 1 (STATIC CORRESPONDENCE). $\Gamma \vdash P : \circ \equiv_{\pi} \Gamma \vdash P' : \circ$ if and only if $\llbracket \Gamma \vdash P : \circ \rrbracket = \llbracket \Gamma \vdash P' : \circ \rrbracket$

Since sorting is preserved by tensor product, and since the addition of a name x can be obtained by tensoring with an idle x , weakening of sorting environments follows for free. Also narrowing comes for free, as sorting is preserved by composition, and narrowing can be expressed by composing with a substitution. In the following, we write a_{π} to denote the π -calculus process – unique up-to structural congruence – corresponding to the agent a .

THEOREM 5 (DYNAMIC CORRESPONDENCE). For every well-typed process $\Gamma \vdash P : \circ$ and agent $a : \epsilon \rightarrow \langle \Gamma \rangle$ we have

$$\llbracket \Gamma \vdash P : \circ \rrbracket \rightarrow a \text{ if and only if } P \rightarrow_{\pi} a_{\pi}$$

The dynamic correspondence between the polyadic π -calculus and its presentation as a bigraphical reactive system has been achieved by working in *soft abstract* bigraphs. In order to examine the behavioural theory it would be convenient to use *hard concrete* bigraphs for their notions of engagedness and RPOs, and then transfer the results back into soft abstract bigraphs.

We let $\mathcal{SBBG}_{\pi \leq} \stackrel{\text{def}}{=} \mathcal{SBBG}_{\text{H}}(\Sigma_{\pi \leq}^{\Delta}, \mathcal{R}_{\pi \leq})$ be the preimage of $\mathbb{SBBG}_{\pi \leq}$ under the quotient functor $\llbracket \cdot \rrbracket_{\Delta}$, so that every rule in $\mathcal{R}_{\pi \leq}$ is a lean preimage of a rule in $\mathcal{R}_{\pi \leq}$. This makes $\mathcal{SBBG}_{\pi \leq}$ a hard concrete

sorted BRS. As the sorting in $\mathcal{SBBG}_{\pi \leq}$ is an instance of subsorting we obtain the following results, relating ST and FPE.

COROLLARY 2.

1. In $\mathcal{SBBG}_{\pi \leq}$ the bisimilarity \sim_{FPE} is a congruence.
2. $a \sim_{\text{FPE}} b$ in $\mathcal{SBBG}_{\pi \leq}$ if and only if $\llbracket a \rrbracket_{\Delta} \sim_{\text{FPE}} \llbracket b \rrbracket_{\Delta}$ in $\mathbb{SBBG}_{\pi \leq}$.
3. The bisimilarity \sim_{FPE} is a congruence in $\mathbb{SBBG}_{\pi \leq}$.

The bisimilarity \sim_{FPE} in $\mathbb{SBBG}_{\pi \leq}$ in turn induces an equivalence \sim_{ind} in $\mathcal{P}_{\pi/\alpha}$. This is the smallest relation on well-sorted $\mathcal{P}_{\pi/\alpha}$ processes relating a_{π} and b_{π} , in symbols $\Gamma \vdash a_{\pi} \sim_{\text{ind}} b_{\pi} : \circ$, whenever a and b are related by \sim_{FPE} and have outer face $\langle \Gamma \rangle$.

THEOREM 6. The relation \sim_{ind} is a congruence.

We follow the same procedure as in [8] to derive transitions, with an extension to ensure well-sortedness. As mentioned above, due to the subsorting and the way the reaction rules are specified, the transitions we obtain may contain redundant subsorting in their labels. However, we prove that we can limit our attention to a subset of labels which do not contain redundant subsorting. We start by characterising the engaged transitions generated by the RPO construction in $\mathbb{SBBG}_{\pi \leq}$.

LEMMA 3 (CHARACTERISING TRANSITIONS IN $\mathbb{SBBG}_{\pi \leq}$). Let $a : \langle \text{sort} \rangle$ be an agent in $\mathbb{SBBG}_{\pi \leq}$ and $a \xrightarrow{L} a'$ one of its engaged transitions. We can characterise a , L , and a' in the following forms

$$\begin{aligned}
a &= (/Z : \bar{S})(r_a \mid b) \\
L &= \langle \sigma \rangle \mid r_L : \langle \text{sort} \rangle \rightarrow \langle \text{sort}' \rangle \\
a' &= \sigma(/Z : \bar{S})(y_1 \dots y_n / (z_1 \dots z_n) c_2 \mid c_1 \mid b) : \langle \text{sort}' \rangle
\end{aligned}$$

where, up to a bijection on names, one of the cases in Fig. 7 holds (ignore the \bar{L} for the moment). In the first (resp. second) case we require that $x \in X$ and that c_2 (resp. c_1 , respectively) is discrete with names not in $X \cup Z$. In the third case we require that $x_0, x_1 \in X$ and that they have a common subsort.

In all the cases the substitution σ can introduce subsorting on the names in X (or, in the third case, on the names in $X \setminus \{x_i\}$).

Furthermore in the first case the sort of x is a subsort of $\text{sort}(x) \sqcap (S_1, \dots, S_n)^t$, where S_1, \dots, S_n are the sort of the edges pointed to by z_1, \dots, z_n . In the second case the sort of x must be a subsort of $\text{sort}(x) \sqcap (\text{sort}'(y_1), \dots, \text{sort}'(y_n))^w$. Finally, in the third case the sort of x_i must be a subsort of $\text{sort}(x_0) \sqcap \text{sort}(x_1)$.

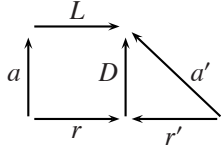
So for a transition $a \xrightarrow{L} a'$, the label L may subsort some of a' 's outer names unnecessarily, as the only name that might need subsorting for the transition to occur is the name communicated over. The reason for the redundant subsorting resides in two related points, which both lead to generate ground reaction rules that could be derived from other rules by subsumption. Firstly, the parametric reaction rules must be indexed on all possible sorts U_i and T_i , due to the requirement on the well-formedness of b-tag sorts, and since the sorts of the outer names y_i depend on those of the edges connected

r_a	r_L	σ	\bar{L}
send $_{xy_1 \dots y_n} c_1$	get $_{x(z_1 \dots z_n)} c_2$	sub $_X$	$x(z_1 : S_1, \dots, z_n : S_n) c_{2\pi}$
get $_{x(z_1 \dots z_n)} c_2$	send $_{xy_1 \dots y_n} c_1$	sub $_X$	$\bar{x}\langle y_1, \dots, y_n \rangle c_{1\pi}$
send $_{x_0 y_1 \dots y_n} c_1 \mid$ get $_{x_1(z_1 \dots z_n)} c_2$	1	sub $_{X \setminus \{x_i\}} \mid x_i/x_{\bar{i}}$	$x_i/x_{\bar{i}}$
send $_{xy_1 \dots y_n} c_1 \mid$ get $_{x(z_1 \dots z_n)} c_2$	1	sub $_X$	τ

Figure 7. Characterisation of transitions

to the inner names z_i . Secondly, the generation of ground reaction rules from parametric ones also adds redundant subsorting. This happens as the grounding of reaction rules maps the parameter's names (and their sort) through without requiring a minimal-sorting condition. There appears to be no easy fix, as the current sorting does not allow to 'minimal-sort' all bigraphs.

These additional ground reaction rules again affect the derivation of transitions. The IPO property of a transition, $a \xrightarrow{L} a'$,



enforces that the label L is *minimal* with respect to the given agent a and the chosen ground reaction rule r . Having an infinite number of reaction rules, we have no elegant way to ensure that we only use the *minimal* reaction rule (among those that only differ on the sorts in their outer face), i.e. the one that introduces the least subsorting when deriving the label.

In order to examine a specific set of reaction rules, we define a sub-lts of FPE, denoted PRE, which has the same agents as FPE but only those transitions from FPE that do not introduce redundant subsorting. We then prove a proposition stating that if PRE and FPE satisfies some simple conditions, then their bisimilarity relations coincide. Following existing terminology, we say that a well-sorted bigraph is a *resorting* if its image under the sorting functor is an identity, and we let φ range over resortings. We say that a label L introduces a redundant subsort if it can be obtained from another label M and a non-identity resorting φ as in the first bullet item of the following proposition.

PROPOSITION 2. *Let \mathcal{L} be a transition system and \mathcal{M} a sub-lts thereof with the following three properties*

- If $a \xrightarrow{L} a'$ is a transition in \mathcal{L} , then there exists a resorting φ and a transition $a \xrightarrow{M} a''$ in \mathcal{M} such that $L = \varphi M$ and $a' = \varphi a''$.
- If $a \xrightarrow{M} a'$ is a transition in \mathcal{L} , then for any resorting φ with the appropriate inner face there exists a transition $a \xrightarrow{\varphi M} \varphi a'$ in \mathcal{L} .
- $\varphi a \xrightarrow{L} a'$ is a transition in \mathcal{L} if and only if $a \xrightarrow{L\varphi} a'$ is a transition in \mathcal{L} .

Then \mathcal{M} is adequate for \mathcal{L} .

We now consider a subset of transitions that introduce no redundant subsorts. This is a subset of the labels of Lemma 3, where the substitution σ assign to x the sort $\text{sort}(x) \sqcap (S_1, \dots, S_n)^f$ in sort' , in the first case, and $\text{sort}(x) \sqcap (\text{sort}'(y_1), \dots, \text{sort}'(y_n))^w$, in the second one. In the third case x is assigned sort $\text{sort}(x_0) \sqcap \text{sort}(x_1)$. We call such transitions *precise*, and we let PRE denote the resulting transition system and \sim_{pre} be the corresponding bisimulation. As PRE and FPE satisfy the conditions of Proposition 2 we obtain the following.

COROLLARY 3. PRE is adequate for FPE.

We can derive a characterisation of the labels for $\mathcal{P}_{\pi/\alpha}$ from the one obtained in Lemma 3 for engaged transitions. In the definition below, for L an engaged label in $\text{SBBG}_{\pi \leq}$, label \bar{L} refers to the corresponding cell of the fourth column in Fig. 7.

DEFINITION 9 (INDUCED TRANSITIONS). The induced transition relation $\xrightarrow{\text{ind}}$ is the smallest relation on typing judgements such that $\Gamma \vdash P : \circ \xrightarrow{\bar{L}}_{\text{ind}} \Gamma' \vdash P' : \circ$ if $\llbracket \Gamma \vdash P : \circ \rrbracket \xrightarrow{L} \llbracket \Gamma' \vdash P' : \circ \rrbracket$ is a precise engaged transition in $\text{SBBG}_{\pi \leq}$.

We now can prove that such labels yield a coinductive characterisation of the induced equivalence.

LEMMA 4. $\Gamma \vdash P \sim_{\text{ind}} Q : \circ$ if and only if $\Gamma \vdash P \mathcal{R} Q : \circ$ for some $\xrightarrow{\text{ind}}$ -bisimulation \mathcal{R} .

Hence, we have identified a suitably small set of labels using the framework of bigraphical reactive system which corresponds to the expected set of labels. And we have proved that the derived labelled transition system characterises the induced congruence.

6. Conclusion and future work

In this paper we have used link-sortings to represent the capability types of Pierce and Sangiorgi in the setting of bigraphs. To this end we have conservatively extended the definition of link graphs to include edge controls, and extended the notion of link-sortings to binding bigraphs. We have proven that the theory developed in [8] can be lifted to a sorted setting with only minor adjustments. Hence, we are able to derive a labelled transition system and a congruential behavioural equivalence for the polyadic π -calculus with capability types. We have furthermore proved that we can confine attention to labelled transitions that do not introduce any redundant subsorting of the agent involved in the transition; indeed, this smaller set of labels induce the same congruence as the full set of labels.

These results are promising, since they provide a foundation for presenting many of the existing type systems for the (polyadic) π -calculus and for similar calculi as sortings in a bigraphical reactive framework. The advantage of that resides of course in the recasting of very successful, yet often incompatible theories, in the uniform, unifying behavioural theory of bigraphical reactive systems, as well as the development and furthering of the foundational framework itself. Technically, we remark that our slight generalisation from 'reflects pushouts' to 'weakly reflects pushouts' allows us to present a wider variety of sortings while enjoying all the benefits of the framework of engaged transitions to derive labelled transitions.

Future work. An obvious path for further work is to examine more advanced type systems presented as sortings in bigraphical reactive systems. We conjecture that our work here can be extended without major changes to the theory to account also for (infinite) recursive types and linear types. However, we expect that to accommodate behavioural information such as channel usage [10] would require a more thorough examination.

We have not yet examined how the derived congruence relates to the traditional equivalences for the polyadic π -calculus. In particular, it would be important to compare the congruence with the typed bisimilarity of [4, 3].

The subsorting presented in the paper relies crucially on the existence of bounded meets, that we use to construct RPOs and to (weakly) reflect pushouts. The requirement of bounded meets is only necessary since we consider a type system with subsorts. It would be interesting to examine more general sufficient conditions on type systems and their features that allow to present them as sortings in a bigraphical reactive system.

Bigraphs are by definition typed with an outer and an inner face; concrete bigraphs in addition have support sets associated to them. So far we have only assigned sorts to interfaces and controls, and not to the bigraph as a whole, as we traditionally do when we consider programming languages, as for instance the λ -calculus, or some process calculi. We plan to examine sortings that assign sorts to and express properties of whole bigraphs, not just their components.

Acknowledgements. For numerous helpful discussions and correspondence we are indebted to Søren Debois, Shane O’Conchuir, Ole Jensen, James Leifer, and last but not least Robin Milner.

References

- [1] L. BIRKEDAL, S. DEBOIS, E. ELSBORG, T. HILDEBRANDT, AND H. NISS. Bigraphical models of context-aware systems. In *Proc. of FOSSACS’06*, volume 3921 of *LNCS*, pages 187–201. Springer 2006.
- [2] M. BUNDEGAARD AND V. SASSONE. Typed polyadic π -calculus in bigraphs. To appear as Technical Report, manuscript available on request, 2006.
- [3] Y. DENG AND D. SANGIORGI. Towards an algebraic theory of typed mobile processes. *Theoretical Computer Science*, 350(2–3):188–212, 2006.
- [4] M. HENNESSY AND J. RATHKE. Typed behavioural equivalences for processes in the presence of subtyping. *Mathematical Structures in Computer Science*, 14(5):651–684, 2004.
- [5] M. HENNESSY AND J. RIELY. Resource access control in systems of mobile agents. *Information and Computation*, 173(1):82–120, 2002.
- [6] O. H. JENSEN. *Mobile Processes in Bigraphs*. PhD thesis, Department of Computer Science, Aalborg University, 2006. Forthcoming.
- [7] O. H. JENSEN AND R. MILNER. Bigraphs and transitions. In *Proc. of POPL’03*, pages 38–49. ACM Press 2003.
- [8] O. H. JENSEN AND R. MILNER. Bigraphs and mobile processes (revised). Technical Report UCAM-CL-TR-580, University of Cambridge, Computer Laboratory 2004.
- [9] B. KLIN, V. SASSONE AND P. SOBOCIŃSKI. Labels from Reductions: Towards a General Theory. In *First International Conference on Algebras and Coalgebras in Computer Science*, volume 3629 of *LNCS*, pages 30–50, Springer 2005.
- [10] N. KOBAYASHI. Type systems for concurrent programs. In *Proc. of 10th Anniversary Colloquium of UNU/IIST*, volume 2757 of *LNCS*, pages 439–453, Springer 2003.
- [11] B. KÖNIG. A general framework for types in graph rewriting. In *Proc. of FSTTCS’00*, volume 1974 of *LNCS*, pages 373–384. Springer 2000.
- [12] B. KÖNIG. A graph rewriting semantics for the polyadic pi-calculus. In *Proc. of GT-VMT’00*, pages 451–458. Carleton Scientific 2000.
- [13] J. J. LEIFER AND R. MILNER. Transition systems, link graphs and Petri nets. Technical Report UCAM-CL-TR-598, University of Cambridge, Computer Laboratory 2004.
- [14] R. MILNER. The polyadic π -calculus: A tutorial. In *Logic and Algebra of Specification*, pages 203–246, Springer 1993.
- [15] R. MILNER. Pure bigraphs: Structure and dynamics. *Information and Computation*, 204(1):60–122, 2006.
- [16] S. O’CONCHUIR. Kind bigraphs – static theory. Technical Report TCD-CS-2005-36, Trinity College Dublin, Computer Science Department 2005.
- [17] B. C. PIERCE AND D. SANGIORGI. Typing and subtyping for mobile processes. *Mathematical Structures in Computer Science*, 6(5):409–453, 1996.
- [18] V. SASSONE AND P. SOBOCIŃSKI. Deriving Bisimulation Congruences using 2-Categories. *Nordic Journal of Computing* vol. 10(2), 163–183, 2003.
- [19] V. SASSONE AND P. SOBOCIŃSKI. Locating Reactions using 2-Categories. *Theoretical Computer Science* vol. 333(1-2), 297–327, 2005.
- [20] P. SEWELL. Global/local subtyping and capability inference for a distributed pi-calculus. In *Proc. of ICALP’98*, volume 1443 of *LNCS*, pages 695–706. Springer 1998.

A. Binding bigraphs, formally

This section briefly recalls the main concepts of binding bigraphs, starting from their constituting place and link graph. We use \uplus for the union of sets known or assumed to be disjoint, and for the union of functions where the domains are such. We define the *support* of a place graph to be the node set, and the support of a link graph and a pure bigraph to be both the node set and the edge set. We define bigraphs and more generally bigraphical reactive systems with respect to a signature.

DEFINITION 10 (SIGNATURE). A *signature* \mathcal{K} is a set whose elements are called *controls*. Each control $K \in \mathcal{K}$ is assigned an arity $ar(K)$, that determines its number of ports. A control can be *atomic*, and then not allowed to contain further structure. Non-atomic controls can be *active*. Controls which are not active are called *passive*.

We begin by defining the two constituents of a pure bigraph: the place graph and the link graph.

DEFINITION 11 (PLACE GRAPH). A *place graph* over a signature \mathcal{K} with *inner* width m and *outer* width n , both finite ordinals, is a structure $G^P = (V, ctrl, prnt)$: $m \rightarrow n$ consisting of a finite set of *nodes* V ; a *control* map $ctrl: V \rightarrow \mathcal{K}$, which assigns controls to nodes; and a *parent* map $prnt: m \uplus V \rightarrow V \uplus n$, which assigns each node and site to parent node or root. The parent map is *acyclic*, i.e. $prnt^k(v) \neq v$ for all $k > 0$ and $v \in V$. An *atomic* node may not be a parent.

The widths m and n index respectively the *sites* and *roots* of A respectively. Sites and nodes are called *places*. We write $w >_A w'$, or just $w > w'$, to indicate that $w = prnt^k(w')$, for some $k > 0$.

DEFINITION 12 (COMPOSITION OF PLACE GRAPHS). The composition of two place graphs $A_i = (V_i, ctrl_i, prnt_i)$: $m_i \rightarrow m_{i+1}$ is defined when A_0 and A_1 have disjoint support. Then the composition $A_1 \circ A_0$ is $(V_0 \uplus V_1, ctrl_0 \uplus ctrl_1, prnt)$: $m_0 \rightarrow m_2$, where $prnt$ is defined as $(id_{V_0} \uplus prnt_1) \circ (prnt_0 \uplus id_{V_1})$.

DEFINITION 13 (TENSOR PRODUCT OF PLACE GRAPHS). The tensor product of place graphs is defined as follows. On widths $m \otimes n$ is the sum of m and n . For two place graphs A_i : $m_i \rightarrow n_i$, their tensor product $A_0 \otimes A_1$: $m_0 \otimes m_1 \rightarrow n_0 \otimes n_1$ is defined when A_0 and A_1 have disjoint support. The tensor product is then defined by ‘shifting’ the sites and roots of A_1 upwards by m_0 and n_0 , respectively, and then taking the union of the two parent maps.

Informally the tensor product of two place graphs is the juxtaposition of the two graphs.

DEFINITION 14 (HARD PLACE GRAPHS). A *hard* place graph is a place graph in which every root and non-atomic node has a child. Ob-

serve that they are closed under both composition and tensor product.

The definition of link graphs follows the pattern of place graphs.

DEFINITION 15 (LINK GRAPH). A *link graph* over a signature \mathcal{K} with finite sets X and Y of respectively of *inner* and *outer* names is a structure $G^\perp = (V, E, ctrl, link): X \rightarrow Y$ consisting of a finite set of *nodes* V ; a finite set of *edges* E ; a *control* map $ctrl: V \rightarrow \mathcal{K}$; and a *link* map $link: X \uplus P \rightarrow E \uplus Y$, where $P \stackrel{\text{def}}{=} \sum_{v \in V} ar(ctrl(v))$ is called the set of *ports* of G^\perp .

The inner names X and the ports P are collectively referred to as the *points* of G^\perp , the edges E and outer names Y as 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. A link is *open* if it is an (outer) name; it is *closed* if it is an edge. A point is *open* if its (image under) the link map is open, otherwise it is *closed*.

DEFINITION 16 (COMPOSITION OF LINK GRAPHS). The composition of two link graphs $A_i = (V_i, E_i, ctrl_i, link_i): X_i \rightarrow X_{i+1}$ is defined when A_0 and A_1 have disjoint supports. Then the composition $A_1 \circ A_0$ is $(V_0 \uplus V_1, E_0 \uplus E_1, ctrl_0 \uplus ctrl_1, link): X_0 \rightarrow X_2$, where *link* is defined as follows. For every $p \in X_0 \uplus P_0 \uplus P_1$, where P_i is the set of ports of A_i

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

DEFINITION 17 (TENSOR PRODUCT OF LINK GRAPHS). The tensor product of link graphs is defined as follows. On name sets $X \otimes Y$ is defined as the union, whenever X and Y are disjoint. For two link graphs $A_i: X_i \rightarrow Y_i$, their tensor product $A_0 \otimes A_1: X_0 \otimes X_1 \rightarrow Y_0 \otimes Y_1$ is defined when both interface products are defined and when A_0 and A_1 have disjoint support. The tensor product is then the union of the two link maps.

DEFINITION 18 (PURE BIGRAPHS). A *pure bigraph* is the superimposition of a place and a link graph sharing the set of nodes and the control map. Namely, $G = (V, E, ctrl, prnt, link): \langle m, X \rangle \rightarrow \langle n, Y \rangle$ where $G^P = (V, ctrl, prnt): m \rightarrow n$ is a place graph, and $G^\perp = (V, E, ctrl, link): X \rightarrow Y$ is a link graph.

Sometimes we will write a pure bigraph G as $G = \langle G^P, G^\perp \rangle$, a combination of its two constituents. We define the composition and the tensor product of two pure bigraphs in terms of the underlying place and link graph.

DEFINITION 19 (COMPOSITION OF PURE BIGRAPHS). The composition of two pure bigraphs $G_i: I_i \rightarrow I_{i+1}$ is defined when G_0 and G_1 have disjoint support. We define $G_1 \circ G_0$ as $\langle G_1^P \circ G_0^P, G_1^\perp \circ G_0^\perp \rangle: I_0 \rightarrow I_2$.

DEFINITION 20 (TENSOR PRODUCT OF PURE BIGRAPHS). The tensor product of two pure bigraphs is defined as follows. On interfaces, $\langle m, X \rangle \otimes \langle n, Y \rangle$ is $\langle m+n, X \uplus Y \rangle$ when X and Y are disjoint. Given two bigraphs $G_i: I_i \rightarrow J_i$ their tensor product $G_0 \otimes G_1$ is $\langle G_0^P \otimes G_1^P, G_0^\perp \otimes G_1^\perp \rangle: I_0 \otimes I_1 \rightarrow J_0 \otimes J_1$, when the interfaces are defined and G_0 and G_1 have disjoint support.

We now enhance pure bigraphs with a primitive mechanism of scoping and binding for names and edges. The first ingredient is to equip nodes with binding ports; the definition of binding bigraphs will then enforce that all points linked to it must lie inside the node.

DEFINITION 21 (BINDING SIGNATURE). A *binding signature* \mathcal{K} assigns to each control K a pair of finite ordinals, the *binding arity* h and the *free arity* k , written $K: h \rightarrow k$, which determine respectively its number of *binding* and *non-binding* ports. Of course, $ar(K) =$

$h + k$. A control can be *atomic*. Non-atomic controls can be *active*. Controls which are not active are called *passive*. If K is *atomic*, then $h = 0$.

In order to enforce a discipline of containment for links the notion of interface must be extended to also assign names to localities, as below.

DEFINITION 22 (BINDING INTERFACE). A *binding interface* is a tuple $I = \langle m, loc, X \rangle$, where $I^u = \langle m, X \rangle$ is a pure interface and $loc: X \rightarrow m$ is a partial *locality* map which associates names X with sites. If $loc(x) = s$ then x is *located* at s , or is *local* to s . If $loc(x) = \perp$ then x is *global*.

Given an interface $\langle m, loc, X \rangle$, we often denote the locality map loc as a vector \vec{X} of length m of pairwise disjoint subsets of X , corresponding to the *loc*-preimages of each site. The names in X not occurring in this way are of course global. We say that an interface I is *local* (resp. *global*) if all its names are local (resp. *global*).

DEFINITION 23 (BINDING BIGRAPHS). A binding bigraph $G: I \rightarrow J$ consists of an *underlying* pure bigraph $G^u: I^u \rightarrow J^u$ with extra structure as follows. Declare the *binders* of G to be the binding ports of its nodes together with the local names of its outer face J . Then G must satisfy the following:

SCOPE RULE: If p is a binder located at a node or, in the case of local names, at a root w , then every peer p' in the same link as p must be located at a place w' (a site, root or node) such that $w' <_{G^u} w$.

A bigraph is *ground* if its inner interface is $\epsilon = \langle 0, (), \emptyset \rangle$, the unit interface. We remark that bigraphs as defined above are usually referred to as ‘concrete’ bigraphs, as opposed to ‘abstract’ bigraphs than can be obtained by forgetting the identities of nodes and edges via a quotient construction (cf. below).

In order to express sorting disciplines that involve also conditions on edges, we find it convenient to extend the definition of bigraphs with the addition of edge controls. An *edge signature* \mathcal{E} is a set whose elements are called *edge controls*. Edge controls will be ranged over by L . Each bigraph G will therefore be equipped also with an *edge* map $edgectrl_G: E \rightarrow \mathcal{E}$ analogous to the control map $ctrl$. As a matter of notation, since any ambiguity can be solved from the context, in the following we shall simply denote both the ‘control’ and ‘edge’ maps with the symbol $ctrl$.

Concrete binding bigraphs are the arrows of a so-called precategory, that is a category but for a partial composition operation. More precisely, they give rise to a particularly well-behaved precategory, an s-category, which support a suitable quotient construction to translate results from concrete to abstract bigraphs. We refer the reader to [13] for the details. A parallel approach has been proposed in [18, 19, 9] based on the notion of bicategories.

DEFINITION 24. The precategory $\mathcal{B}BG(\mathcal{K}, \mathcal{E})$ of concrete binding bigraphs over signatures \mathcal{K} and \mathcal{E} has binding interfaces as objects and binding bigraphs as arrows. If $G: I \rightarrow J$ and $H: J \rightarrow K$ are concrete binding bigraphs with disjoint support, then their composition is defined directly in terms of composition of the underlying pure bigraphs.

$$H \circ G \stackrel{\text{def}}{=} H^u \circ G^u$$

In the same manner, the identities $\text{id}_I: I \rightarrow I$, where $I = \langle m, loc, X \rangle$ are defined in terms of the underlying identities. We denote by $\mathcal{B}BG_h(\mathcal{K}, \mathcal{E})$ the subprecategory of $\mathcal{B}BG(\mathcal{K}, \mathcal{E})$ consisting of *hard* binding bigraphs.

There exists a forgetful functor from the precategory of binding bigraphs to the precategory of pure bigraphs

$$\mathcal{U}: \mathbf{BBG}(\mathcal{K}, \mathcal{E}) \rightarrow \mathbf{BIG}(\mathcal{K}, \mathcal{E}) ,$$

which sends each interface (and bigraph) to the underlying interface (and bigraph), e.g. I to I^u and G to G^u . Likewise, we have a forgetful functor for hard binding bigraphs to hard pure bigraphs.

PROPOSITION 3. *On interfaces the forgetful functor is surjective (but not in general injective); it is also faithful, i.e. injective (though not in general surjective) on each homset of binding bigraphs.*

DEFINITION 25 (TENSOR PRODUCT). We define the tensor product of two interfaces $I = \langle m, \vec{X}, X \rangle$ and $J = \langle n, \vec{Y}, Y \rangle$, where X and Y are disjoint as

$$I \otimes J = \langle m + n, \vec{X}\vec{Y}, X \uplus Y \rangle .$$

We then define the tensor product of two binding bigraphs $G_i: I_i \rightarrow J_i$ ($i = 0, 1$) with disjoint support in terms of the underlying pure bigraph $G^u = G_0^u \otimes G_1^u$, whenever $I = I_0 \otimes I_1$ and $J = J_0 \otimes J_1$ are defined.

Here we define the most important concepts behind s-categories, we refer the reader to [15] for the full definition of s-categories.

DEFINITION 26 (S-CATEGORY). An *s-category* \mathbf{A} is a strict symmetric monoidal precategory with the following structure:

- for each arrow f we have a finite set $|f|$ called the support of f , such that $|\text{id}_f| = \emptyset$. We define the composition of two arrows f and g iff their domain and codomain match and iff $|g| \cap |f| = \emptyset$, we define the support of their composition as $|gf| = |g| \uplus |f|$. Similar we define the tensor product of two arrows f and g if their support is disjoint and define the support of their product as $|f \otimes g| = |f| \uplus |g|$.
- for any arrow $f: I \rightarrow J$ and any injective map ρ with a domain including $|f|$ we have an arrow $\rho \cdot f: I \rightarrow J$ and satisfying the following rules:
 1. $\rho \cdot \text{id}_I = \text{id}_I$
 2. $\rho \cdot (gf) = (\rho \cdot g)(\rho \cdot f)$
 3. $\rho \cdot (f \otimes g) = \rho \cdot f \otimes \rho \cdot g$
 4. $\text{id}_{|f|} \cdot f = f$
 5. $(\rho_1 \circ \rho_0) \cdot f = \rho_1 \cdot (\rho_0 \cdot f)$
 6. $\rho \cdot f = (\rho \upharpoonright |f|) \cdot f$
 7. $|\rho \cdot f| = \rho(|f|)$,

where $\rho \upharpoonright |f|$ is the function ρ restricted to the support of f . All the equations are required to hold only when both sides are defined.

An s-category \mathbf{A} is wide if it is equipped with a functor $\text{width}: \mathbf{A} \rightarrow \mathbf{Ord}$, the category of finite ordinals, such that $\text{width}(\epsilon) = 0$ and each bijection π on $\text{width}(I)$ is the image of an isomorphism $\pi_I: I \rightarrow I$ in \mathbf{A} .

THEOREM 7. *For any node signature \mathcal{K} and edge signature \mathcal{E} the precategories $\mathbf{BBG}(\mathcal{K}, \mathcal{E})$ and $\mathbf{BBG}_h(\mathcal{K}, \mathcal{E})$ are wide s-categories, with the origin $\epsilon = \langle 0, (\emptyset), \emptyset \rangle$ and where the interface $\langle m, \text{loc}, X \rangle$ has $\text{width}(I) = m$ and for a bigraph G the width map $\text{width}(G)$ sends each site s in G to the unique root r such that $s <_G r$.*

A link is *bound* if it contains a binder, otherwise it is *free*.

DEFINITION 27 (DISCRETENESS, LEANNESS). A binding bigraph is *discrete* if every free link is a name and has exactly one point. A binding bigraph is *lean* if it has no idle edges.

There exists a ‘lean-support equivalence’ quotient functor from concrete to abstract binding bigraphs.

DEFINITION 28 (ABSTRACT BINDING BIGRAPHS). An *abstract binding bigraph* is a lean-support equivalence class of concrete binding bigraphs. For any signatures \mathcal{K} and \mathcal{E} we have the category

$\mathbf{BBG}(\mathcal{K}, \mathcal{E})$ and the quotient functor $\llbracket \cdot \rrbracket: \mathbf{BBG}(\mathcal{K}, \mathcal{E}) \rightarrow \mathbf{BBG}(\mathcal{K}, \mathcal{E})$, and similarly for hard binding bigraphs.

DEFINITION 29 (WIRING, CLOSURE, SUBSTITUTION). A bigraph with interfaces of zero width is called a *wiring*. Since the interfaces are of zero width the bigraph cannot have nodes, and takes the form $\omega: X \rightarrow Y$. There are two basic forms of wirings, closures and substitutions. A *closure* $(/x: C): x \rightarrow \epsilon$ maps the inner name x to the single edge (with control C) that the bigraph contains.

A *substitution* $\sigma: X \rightarrow Y$ is a function that is not necessarily surjective. We denote by $x: \epsilon \rightarrow x$ the empty substitution from ϵ to x . We write \vec{y}/\vec{x} , whenever \vec{y} and \vec{x} are vectors of equal length, to denote the surjective substitution $x_i \mapsto y_i$. Both closure and substitution can be generalised to an arbitrary set of names by tensoring together one operation per name in the set.

DEFINITION 30 (PRIME BIGRAPH). An interface is *prime* if it has width 1. A binding bigraph $G: I \rightarrow J$ is *prime*, if I is local and J is prime.

DEFINITION 31 (CONCRETION, ABSTRACTION). We define a *concretion* as a prime $\ulcorner X \urcorner: \langle 1, (X \uplus Y), X \uplus Y \rangle \rightarrow \langle 1, (Y), X \uplus Y \rangle$, which globalises a subset if its local inner names. Dual to the concretion we define the *abstraction* on a prime P , which localises a subset of the global names of P . So for $X \uplus Y \subseteq Z$ and prime $P: I \rightarrow \langle 1, (Y), Z \rangle$ we may create the abstraction $(X)P: I \rightarrow \langle 1, (X \uplus Y), Z \rangle$. Since these notions are dual, we have the following

$$(\ulcorner X \urcorner \otimes \text{id}) \circ (X)P = P \text{ and } (X)\ulcorner X \uplus Y \urcorner = \ulcorner Y \urcorner .$$

Also we can define *local wirings* which work on local names. We define the *closure* $(/X)$ as $(/X \otimes \text{id}) \circ \ulcorner X \urcorner$; the *substitution* $\vec{y}/(\vec{x})$ as $(\vec{y}/\vec{x} \otimes \text{id}) \circ \ulcorner \vec{x} \urcorner$; and the *local substitution* $(\vec{y})/(\vec{x})$ as $(\vec{y})\vec{y}/(\vec{x})$.

DEFINITION 32 (IONS, ATOMS, AND MOLECULES). For a non-atomic control $K: h \rightarrow k$, let \vec{x} and \vec{y} be sequences of distinct names of length h and k . Let $X = \{\vec{x}\}$, $Y = \{\vec{y}\}$ and define the *free discrete ion* $K_{\vec{y}/\vec{x}}: \langle 1, (X), X \rangle \rightarrow \langle 1, (\emptyset), Y \rangle$ to have local inner names \vec{x} and global outer names \vec{y} linked to respectively the h binding and the k non-binding ports of a single K -node. For any prime discrete P with outer face $\langle 1, (X), X \uplus Z \rangle$ we call $(K_{\vec{y}/\vec{x}} \otimes \text{id}_Z) \circ P$ a *free discrete molecule*, its outer face is $\langle 1, (\emptyset), Y \uplus Z \rangle$. For atomic K a *free discrete atom* is just $K_{\vec{y}}: \epsilon \rightarrow \langle 1, (\emptyset), Y \rangle$.

DEFINITION 33 (PARALLEL PRODUCT). The *parallel product* of two interfaces $J_i = \langle n_i, \vec{X}_i, Y_i \rangle$ ($i = 0, 1$) is defined, when the local names of each are disjoint from all names of the other, i.e. $Z_0 \cap Y_1 = Z_1 \cap Y_0 = \emptyset$, where $Z_i = \uplus \{\vec{X}_i\}$ ($i = 0, 1$), as

$$J_0 \parallel J_1 \stackrel{\text{def}}{=} \langle n_0 + n_1, \vec{X}_0 \vec{X}_1, Y_0 \cup Y_1 \rangle .$$

Let $G_i = I_i \rightarrow J_i$ ($i = 0, 1$) be two binding bigraphs with disjoint support and where $I_0 \otimes I_1$ and $J_0 \parallel J_1$ are defined, then their *parallel product* is

$$G_0 \parallel G_1 \stackrel{\text{def}}{=} \sigma(G_0 \otimes \tau G_1): I_0 \otimes I_1 \rightarrow J_0 \parallel J_1 ,$$

where the substitutions σ and τ involve only global names and are defined as follows: z_i ($i \in n$) are the names that are shared between G_0 and G_1 , and w_i are fresh names in bijection with the z_i , then $\tau(z_i) = w_i$ and $\sigma(w_i) = \sigma(z_i) = z_i$ ($i \in n$).

DEFINITION 34 (PRIME PRODUCT). The *prime product* of two interfaces $J = \langle n, \vec{X}, X \rangle$ and $I = \langle m, \vec{Y}, Y \rangle$ is defined, when the local names of each are disjoint from all names of the other, as

$$J \mid I \stackrel{\text{def}}{=} \langle 1, (Z_0 \uplus Z_1), X \cup Y \rangle$$

where $Z_0 = X_0 \uplus \dots \uplus X_{n-1}$ and $Z_1 = Y_0 \uplus \dots \uplus Y_{m-1}$.

Let $G_i = I_i \rightarrow J_i$ ($i = 0, 1$) be two binding bigraphs with disjoint support and where $I_0 \otimes I_1$ is defined and n is the sum of the widths

of J_0 and J_1 , then the *prime product* is

$$G_0 | G_1 \stackrel{\text{def}}{=} \text{merge}_n \circ (G_0 \parallel G_1): I_0 \otimes I_1 \rightarrow J_0 | J_1 ,$$

where merge_n is defined as a bigraph with no nodes nor edges, which maps the n sites to a single root with the appropriate mapping of names.

DEFINITION 35 (INSTANTIATION). We define an *instantiation* ϱ from $I = \langle m, \vec{X}, X \rangle$ to $J = \langle n, \vec{Y}, Y \rangle$ (both interfaces local), written $\varrho :: I \rightarrow J$, in terms of an underlying function $\bar{\varrho}: n \rightarrow m$. For any set of names Z this function defines the following map between ground bigraphs

$$\varrho: (I \otimes Z) \rightarrow (J \otimes Z) .$$

First decompose $g: I \otimes Z$ into $g = \omega(d_0 \otimes \cdots \otimes d_{m-1})$ with $\omega: W \rightarrow Z$ a wiring and each d_i prime and discrete. Then define

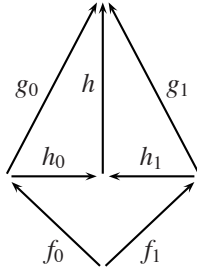
$$\varrho(g) \stackrel{\text{def}}{=} \omega(e_0 \parallel \cdots \parallel e_{n-1}) ,$$

where $e_j \simeq d_{\bar{\varrho}(j)}$, for $j \in n$. An instantiation is *affine* whenever the underlying function $\bar{\varrho}$ is injective, meaning that the instantiation does not replicate any of its parameters.

B. RPOs and IPOs

In this section we define the notions of relative pushout and idem pushout, as stated in [8], and the most important definitions and results regarding binding bigraphs.

DEFINITION 36 (RELATIVE PUSHOUT). Suppose that the outer diagram commutes. We say that (h_0, h_1, h) is a bound for f_i relative to g_i if h_i is a bound for f_i and $h \circ h_i = g_i$. Such a triple is universal if for any other bound (k_0, k_1, k) for f_i relative to g_i there exists a unique mediating arrow j such that $j \circ h_i = k_i$ and $k \circ j = h$. In such a case, we will call the triple a *relative pushout* (RPO).



DEFINITION 37 (IDEM PUSHOUT). Given a span f_i , a bound h_i is an *idem pushout* (IPO) for f_i if the triple (h_0, h_1, id) is an RPO for f_i to h_i .

DEFINITION 38 (REACTION RULES FOR BINDING BIGRAPHS). A *ground reaction rule* is a pair (r, r') , where r and r' are ground with the same outer face. Given a set of ground rules, the reaction relation \rightarrow over agents is the least relation, closed under support equivalence (\simeq), such that $D \circ r \rightarrow D \circ r'$ for each active context D and each ground rule (r, r') .

A *parametric (reaction) rule* has a *redex* R and a *reactum* R' , and takes the following form

$$(R: I \rightarrow J, R': I' \rightarrow J, \varrho)$$

where the inner faces I and I' are local with widths m and m' , and $\varrho :: I \rightarrow I'$ is an affine instantiation, as defined in Definition 35. For every set of names X and discrete parameter $d: \epsilon \rightarrow X \otimes I$ the parametric rule generates the ground reaction rule

$$((\text{id}_X \otimes R) \circ d, (\text{id}_X \otimes R') \circ \varrho(d)) .$$

We define a bigraphical reactive system (BRS) as in [8, Definition 12.2], where the following results were proved.

DEFINITION 39 (BIGRAPHICAL REACTIVE SYSTEM). We define a bigraphical reactive system (BRS) over signatures \mathcal{K} and \mathcal{E} to consist of $\mathcal{B}BG(\mathcal{K}, \mathcal{E})$ and a set of reaction rules \mathcal{R} closed under support equivalence. We then write $\mathcal{B}BG(\mathcal{K}, \mathcal{E}, \mathcal{R})$ for the bigraphical reactive system.

COROLLARY 4 (CONGRUENCE OF WIDE BISIMILARITY). *In any concrete BRS equipped with the standard transition system st , wide bisimilarity \sim of agents is a congruence.*

PROPOSITION 4 (TRANSITIONS RESPECT EQUIVALENCE). *In any concrete BRS with all redexes lean, equipped with st :*

1. *In every transition label L , both components are lean.*
2. *Transitions respect lean-support equivalence (\simeq). That is, for every transition $a \xrightarrow{L} a'$, if $a \simeq b$ and $L \simeq M$, where M is another label with $M \circ b$ defined, then there exists a transition $b \xrightarrow{M} b'$ for some b' such that $a' \simeq b'$.*

COROLLARY 5 (BEHAVIOURAL CONGRUENCE IN ABSTRACT BRS). *Let \mathbf{A} be a concrete BRS with all redexes lean, equipped with st , and let $\llbracket \cdot \rrbracket: \mathbf{A} \rightarrow \mathbf{A}$ be the quotient functor by lean-support equivalence. Then*

1. *$a \sim b$ in \mathbf{A} if and only if $\llbracket a \rrbracket \sim \llbracket b \rrbracket$ in \mathbf{A} .*
2. *Bisimilarity is a congruence in \mathbf{A} .*