Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science 225 (2009) 361–377
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Kind Bigraphs

Shane Ó Conchúir[1,2]

School of Computer Science and Statistics
Trinity College Dublin
Dublin, Ireland

Abstract

We present a refinement, suggested by Jensen and Milner under the term kind, of pure bigraphs. We name the result kind bigraphs. This refinement generalises the notion of atomic and non-atomic controls, allowing a control to contain a subset of the set of controls.
We show that this variation has relative pushouts and classify its idem pushouts. A canonical labelled transition system can be derived from this classification and we use known results to reason about bisimilarity on this transition system. We show how kind bigraphs can be used to describe Milner's homomorphic sortings and finally discuss the extra expressivity that parametric kind reaction rules allow.

Keywords: bigraphs, reactive systems, sorting, relative pushouts

1 Introduction

Bigraphs [7,10,11] are a framework for modelling mobile, distributed agents with connectivity similar to π-calculus terms and locality similar to mobile ambients. A bigraph of a given bigraphical reactive system (Brs) can be compared to a process calculus term. A bigraph consists of a hierarchy of nodes called *controls* which may be linked together or to names of the bigraph. A category of bigraphs equipped with a set of reaction rules which allows the bigraphs to reconfigure themselves forms a Brs. Many standard calculi can be modelled in this framework; the asynchronous π-calculus [8], mobile ambients [6], condition-event Petri nets [10], arithmetic nets [10], finite CCS [11], and a λ-calculus with explicit substitution [12].

Bigraphs are presented using category theory. A Brs is defined as a (pre)category over a signature of controls. Bigraphs are morphisms of the category where a composition $G \circ F$ represents filling the holes of a context bigraph G with a bigraph

[1] This research was partially supported by the Irish Research Council for Science, Engineering, and Technology: funded by the National Development Plan.

[2] Email:Shane.OConchuir@cs.tcd.ie

doi:10.1016/j.entcs.2008.12.086

F. The operational semantics of a Brs is given as a labelled transition system (LTS) which is canonically derived from category-theoretic universal constructions called idem pushouts (IPOs).

The theory of bigraphs is general; the controls of a specific Brs may represent the term constructors of some calculus, physical entities in a smart building, or some abstract concept. The pure theory [11] is quite unconstrained. The two notions under study – mobile locality and mobile connectivity – are represented by two (hence 'bi') almost independent and fairly free structures. Particularly (for this paper), the hierarchical structure is merely two-sorted; a control may either contain any other control (*non-atomic*) or else none at all (*atomic*).

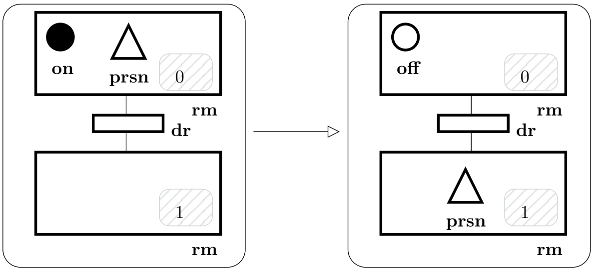
The theory is also extendable. Pure bigraphs have quite a rich and useful theory but, as has both been recognised and studied in the works cited above, some significant applications require refinements of the theory. For example, bigraphical encodings of the ambient calculus and the π-calculus [6] will likely employ a sorting on the term structure to match the grammar of the original calculi and name-scoping is essential to models of the π-calculus [8] and the λ-calculus [12].

Extensions take advantage of the category theoretic foundation. A general method is as follows. One constrains or extends the definition of signature, enriches the objects of the (pre)categories, places suitable constraints/extends the two structures, and possibly weakens the independence between them. Next, a functor is defined from the extended bigraphs to pure bigraphs which forgets these additions. Finally, the functor is shown to have certain properties (*e.g.* faithfulness, preservation/reflection of certain universal constructions) which are used to prove that the good features of pure bigraphs (*e.g.* congruence of bisimilarity, 'canonical' LTSs) are present in the extension. We present such an extension here.

1.1 This paper

A bigraph consists of two graphs which share the same set of controls – a *place graph*, a tree-like structure which captures the idea of locality or containment, and a *link graph* which connects names and controls together. Nodes of different controls are depicted with different shapes, nested according to the place graph, and linked together according to the link graph. Examples of bigraphical rewrite rules are shown in Figure 1 and Figures 2–5.

Jensen and Milner suggested assigning a *kind* to each node of a bigraph, determining the controls of the nodes it may contain. We have formalised their suggestion and proven that the result, *kind bigraphs*, yields a generalisation of pure bigraphs which retains much of the pure theory [14]. Our approach is to refine the notion of atomicity by allowing a control of a signature $\mathcal{K}$ to contain a subset of the set of controls of $\mathcal{K}$. This allows more correct modelling of abstract systems (*e.g.* smart buildings) but also encompasses the sorting used to model finite CCS [11]. Kind bigraphs retain both the RPO constructions necessary to generate the labels of the canonical LTS associated with a bigraphical system and the property that bisimilarity over this LTS is a congruence. As expected, their reaction rules are more expressive than those of pure bigraphs. Interestingly, we will also demonstrate how they may sometimes allow some basic flow control in the reaction relation.



$$/l/r(\mathbf{rm}_l(\mathbf{prsn}|\mathbf{on}|\text{PC})|\mathbf{dr}_{lr}|\mathbf{rm}_r\square) \longrightarrow /l/r(\mathbf{rm}_l(\mathbf{off}|\text{PC})|\mathbf{dr}_{lr}|\mathbf{rm}_r(\mathbf{prsn}|\square))$$

Fig. 1. Lights turn off when the last person exits the room

1.2 Motivation

We are interested in specification. Bigraphs are intended for practical as well as theoretic applications. The bigraphical programming languages project [1] aims to provide a formal language for mobile distributed systems. Similarly, we are interested in modelling abstract systems such as 'smart buildings' where the hierarchical structure reflects physical containment. Physical containment is not simply atomic/non-atomic; physical laws prohibit certain objects being contained inside other objects. We wish to express these constraints on bigraphs so that this structure may be represented in the Brss, allowing finer specification.

Modal logics for process calculi and frameworks [4,3] allow some specifications to be verified in the logics. Conforti, Macedonio, and Sassone [5] have investigated modal logics for bigraphs. All these logics are modal in both time and space and the latter is particularly interesting to us. Our feeling is that the extra structure in kind bigraphs will allow for a modal logic which can meet some specifications that pure bigraphs can not. We will return to this briefly in Section 5.1.

2 Definitions

We refer the reader to the original work [7,11] for a formal definition of the bigraphical machinery. For practical considerations, we can only concentrate on a few aspects required for our presentation and omit some details. Any proofs omitted here may be found in our technical report [14]. The numbered figures in the paper are decorated with algebraic terms which are similar in notation to the presentation in the original work but we do not employ the term structure anywhere in this paper.

We assume a knowledge of basic category theory here. The symbol $\uplus$ denotes the union of two sets which are required to be disjoint, $\boxplus$ denotes the union of two functions whose domain is required to be disjoint, $\circ$ or juxtaposition denotes composition, and id_X denotes the identity arrow on the object X.

Definition 2.1 [dynamic signature [13]] A *dynamic signature* $\mathcal{K}$ $\{\mathcal{K}, ar, status\}$ is composed of a set $\mathcal{K}$ of *controls* and two maps $ar : \mathcal{K} \to \mathbb{N}$ and $status : \mathcal{K} \to \{\mathsf{atomic}, \mathsf{passive}, \mathsf{active}\}$. The ar map assigns an *arity* to a control. The $actv$ function determines which controls are *atomic*. If a control is *passive* or *active* then it is *non-atomic*.

The arity of a control specifies the number of ports each node of that control has. Ports are used to link controls together (*e.g.* to bind a variable node to an abstraction node in a model of the λ-calculus or to represent that some key 'belongs to' some lock in a cryptographic model) or to link controls to free names. Only nodes with non-atomic controls may contain other nodes and so every dynamic signature gives rise to a sorting $kind : \mathcal{K} \to \{\emptyset, \mathcal{K}\}$ specifying whether a node of control K may contain no nodes (atomic) or else contain nodes of any control (non-atomic). Kind signatures generalise this notion.

Definition 2.2 [kind signature, contain] A *kind signature* $\{\mathcal{K}, ar, actv, kind\}$ is composed of a set $\mathcal{K}$ of *controls* and three maps $ar : \mathcal{K} \to \mathbb{N}$, $actv : \mathcal{K} \to \{\mathsf{passive}, \mathsf{active}\}$, and $kind : \mathcal{K} \to \mathcal{P}(\mathcal{K})$ where if $kind(K) = \emptyset$ then $actv(K) = \mathsf{passive}$.

For $K, K' \in \mathcal{K}$, if $K' \in kind(K)$ then we say that a node of control K can *contain* a node of control K' and call $kind(K)$ the *kind* of K.

We typically use $\mathcal{K}$ to denote an arbitrary signature rather than just the set of controls.

Kind bigraphs over a signature $\mathcal{K}$ form an s-category. An s-category is a particular kind of strict symmetric monoidal precategory. *Precategory* suggests that composition is not always defined. *Monoidal* suggests a tensor product; this turns out to be similar to parallel composition in *e.g.* the π-calculus. *Strict* and *symmetric* describe properties of this tensor. The definitions are presented in the appendix.

We first define interfaces, the objects of these s-categories, and then kind bigraphs, the arrows. The latter are presented by separately describing the orthogonal link graph and place graphs components. We presuppose an infinite set $\mathcal{X}$ of *names* in the following.

Definition 2.3 [link graph] A *link graph* $A = (V, E, ctrl, link) : X \to Y$ has finite sets X of *inner names*, Y of *outer names*, V of nodes, and E of edges. It also has a function $ctrl : V \to \mathcal{K}$ called the *control map*, and a function $link : X \uplus P \to E \uplus Y$ called the *link map*, where $P \stackrel{\text{def}}{=} \sum_{v \in V} ar(ctrl(v))$ is the set of *ports* of A.

Definition 2.4 [s-category of link graphs] The s-category $\textsc{Lig}(\mathcal{K})$ of link graphs over $\mathcal{K}$ has name sets as objects and link graphs as arrows. The composition $A_1 \circ A_0 : X_0 \to X_2$ of two link graphs $A_i = (V_i, E_i, ctrl_i, link_i) : X_i \to X_{i+1}$ $(i = 0, 1)$ is defined when their node sets and edge sets are disjoint; then $A_1 \circ A_0 \stackrel{\text{def}}{=} (V, E, ctrl, link)$ where $V = V_0 \uplus V_1$, $ctrl = ctrl_0 \uplus ctrl_1$, $E = E_0 \uplus E_1$, and $link = (\mathrm{id}_{E_0} \uplus link_1) \circ (link_0 \uplus \mathrm{id}_{P_1})$. The identity link graph at X is $\mathrm{id}_X \stackrel{\text{def}}{=} (\emptyset, \emptyset, \emptyset_{\mathcal{K}}, \mathrm{id}_X) : X \to X$.

The tensor product $\otimes$ in $\mathcal{L}$IG is defined on objects as $X \otimes Y \stackrel{\text{def}}{=} X \uplus Y$. The tensor product $A_0 \otimes A_1$ of two link graphs with disjoint nodes and edges is defined when their interface products are defined by taking the union of the link maps.

Definition 2.5 [kind place interface] A *kind place interface* $I = \langle m, \overrightarrow{\theta} \rangle$ over a kind signature is a pair where the *width* m is a finite ordinal and $\overrightarrow{\theta} = (\theta_0, \ldots, \theta_{m-1})$ is a sequence of *kind sorts* where $kind(i) = \theta_i \subseteq \mathcal{P}(\mathcal{K})$ for $i \in m$.

Definition 2.6 [kind place graph] A *kind place graph* $A = (V, ctrl, prnt) : \langle m, \overrightarrow{\theta} \rangle \rightarrow \langle n, \overrightarrow{\theta}' \rangle$ has an *inner width* m and an *outer width* n – both finite ordinals – a finite set V of nodes with a *control map* $ctrl : V \rightarrow \mathcal{K}$, and a *parent map* $prnt : m \uplus V \rightarrow V \uplus n$. The parent map is *acyclic i.e.* $prnt^k(v) \neq v$ for all $k > 0$ and $v \in V$. The widths m and n index the *sites* and *roots* of A respectively and completely. Additionally, A must satisfy the following *kind rules*:

K1 if $p = G(v)$ then $ctrl(v) \in kind(p)$;

K2 if $p = G(s)$ then $kind(s) \subseteq kind(p)$;

K3 if $kind(v) = \emptyset$, v has no children;

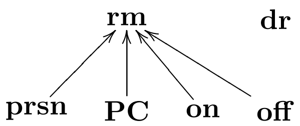
where p is a root or node, s a site, v is a node, and the kind of a node is the kind of its control.

The sites and roots of a place graph are used to compose place graphs and bigraphs (below). The sites of a place graph are akin to the holes of a context in a term calculus.

The containment relationship (the *kind* function) of a kind signature may be represented as a directed acyclic graph. For example, the set of controls

$$\mathcal{K} = \{\mathbf{rm}, \mathbf{dr}, \mathbf{prsn}, \mathbf{PC}, \mathbf{on}, \mathbf{off}\}$$

representing rooms, doors, people, PCs, lit lights, and unlit lights, may be used in a signature modelling an abstract model of buildings and workers[3]. An appropriate nesting structure may then be depicted as below.



where an arc from one control K to another K' states that $K \in kind(K')$, representing 'can be contained in.' The bigraphs in Figure 1 conform to this signature.

Using this graphical representation of *kind*, a kind place graph may be informally described as a forest of unordered trees whose roots and sites are ordered and which respect the containment structure the representation implies. This constraint is only on parent-child relationships – the property 'can be contained in' is not necessarily transitive.

[3] This signature is based on the DELCA example given by the bigraphical programming languages group.

Definition 2.7 [s-category of kind place graphs] The s-category $\mathcal{K}\textsc{pg}(\mathcal{K})$ of kind place graphs over $\mathcal{K}$ has kind interfaces as objects and place graphs as arrows. The composition $A_1 \circ A_0 : m_0 \to m_2$ of two kind place graphs $A_i = (V_i, ctrl_i, prnt_i) : \langle m, \overrightarrow{\theta} \rangle \to \langle m_{i+1}, \overrightarrow{\theta}' \rangle (i = 0, 1)$ with disjoint sets of nodes is $A_1 \circ A_0 \stackrel{\text{def}}{=} (V, ctrl, prnt)$ where $V = V_0 \uplus V_1$, $ctrl = ctrl_0 \uplus ctrl_1$, and $prnt = (\mathrm{id}_{V_0} \uplus prnt_1) \circ (prnt_0 \uplus \mathrm{id}_{V_1})$. The identity place graph at $\langle m, \overrightarrow{\theta} \rangle$ is $\mathrm{id}_{\langle m, \overrightarrow{\theta}' \rangle} \stackrel{\text{def}}{=} (\emptyset, \emptyset_{\mathcal{K}}, \mathrm{id}_m) : \langle m, \overrightarrow{\theta} \rangle \to \langle m, \overrightarrow{\theta} \rangle$.

The *tensor product* $\otimes$ in $\mathcal{K}\textsc{pg}(\mathcal{K})$ is defined on objects as $m \otimes n \stackrel{\text{def}}{=} m + n$. The tensor product $A_0 \otimes A_1$ of two place graphs with disjoint sets of nodes and where $A_0 : m \to n$ is given by defining the parent map by first offsetting the sites and roots of A_1 by m and n respectively, and then taking the union of the two parent maps. Informally, the tensor product of two place graphs places them side-by-side.

Link graphs and kind place graphs are combined to define kind bigraphs.

Definition 2.8 [kind interface] A *kind interface* $I = \langle m, \overrightarrow{\theta}, X \rangle$ over a kind signature consists of a width m, a finite set X of names drawn from $\mathcal{X}$, and a sequence of sorts $\overrightarrow{\theta}$ where each sort is an element of $\mathcal{P}(\mathcal{K})$.

Definition 2.9 [kind bigraph] A *kind bigraph* over a signature $\mathcal{K}$ takes the form $G = (V, E, ctrl, G^{\mathrm{P}}, G^{\mathrm{L}}) : I \to J$ where $I = \langle m, \overrightarrow{\theta}, X \rangle$ and $J = \langle n, \overrightarrow{\theta}', Y \rangle$, its *inner* and *outer faces*, are kind interfaces. Its first two components V and E are finite sets of *nodes* and *edges* respectively. The third component $ctrl : V \to \mathcal{K}$ is a control map. The last two components are a kind place graph $G^{\mathrm{P}} = (V, ctrl, prnt) : \langle m, \overrightarrow{\theta} \rangle \to \langle n, \overrightarrow{\theta}' \rangle$ and a link graph $G^{\mathrm{L}} = (V, E, ctrl, link) : X \to Y$.

Definition 2.10 [s-category of kind bigraphs] The s-category $\mathcal{K}\textsc{bg}(\mathcal{K})$ of kind bigraphs over $\mathcal{K}$ has kind interfaces $I = \langle m, \overrightarrow{\theta}, X \rangle$ as objects and kind bigraphs $G = (V, E, ctrl_G, G^{\mathrm{K}}, G^{\mathrm{L}}) : I \to J$ as arrows. If a bigraph $H : J \to K$ has both its node set and edge set disjoint from V and E respectively, then their composition is defined as

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

The identities are $\langle \mathrm{id}_{\langle m, \overrightarrow{\theta} \rangle}, \mathrm{id}_X \rangle : I \to I$, where $I = \langle m, \overrightarrow{\theta}, X \rangle$. The tensor product of two bigraphs is defined when the tensor product of their link and place graphs are defined.

Informally, composition of kind bigraphs consists of 'planting' one place graph inside another (via the roots and sites) and fusing the link graphs together whereas tensor product places two bigraphs in juxtaposition.

Definition 2.11 [fitting bigraph] A kind bigraph $G : I \to \langle n, \overrightarrow{\theta}, X \rangle$ is said to be *fitting* if each sort $\theta_i, i \in n$ of $\overrightarrow{\theta}'$ is the least subset of $\mathcal{K}$ which satisfies K1 and K2.

Fitting bigraphs over a signature $\mathcal{K}$ form a sub-s-category (see appendix) $\mathcal{F}\textsc{kb}(\mathcal{K})$ of $\mathcal{K}\textsc{bg}(\mathcal{K})$ with particularly strong properties mentioned later.

Proposition 2.12 *Identities, composition, and tensor product in* $\mathcal{K}\textsc{bg}(\mathcal{K})$ *and* $\mathcal{F}\textsc{kb}(\mathcal{K})$ *respect the kind rules. Further, the composition and tensor product of two fitting bigraphs is a fitting bigraph.*

Pure bigraphs have the same structure as kind bigraphs except that: i) the objects of s-categories of pure bigraphs are pairs $\langle m, X\rangle$ without kind sorts; and ii) pure bigraphs do not have to satisfy the kind rules. The identities, composition, and tensor product of pure bigraphs are defined as in Definition 2.10 (forgetting the kind sorts). Kind signatures can be transformed into dynamic signatures by defining $status(K) = actv(K)$ when $kind(K) \neq \emptyset$ and $status(K) =$ atomic otherwise. $`\textsc{Big}(\mathcal{K})$ denotes the s-category of pure bigraphs over the kind signature $\mathcal{K}$. There is a forgetful functor $\mathcal{U}_{\mathcal{K}} : `\textsc{Kbg}(\mathcal{K}) \to `\textsc{Big}(\mathcal{K})$ from kind (resp. fitting) bigraphs to pure bigraphs. It is forgetful in that it forgets about the vectors $\overrightarrow{\theta}$ in the interfaces.

Proposition 2.13 *$\mathcal{U}_{\mathcal{K}}$ is surjective on objects and is faithful.*

Milner introduced place-sortings to describe sortings of the place graph structure of pure bigraphs. In the following definitions, Θ denotes a non-empty set of *sorts*, θ ranges over Θ and $\mathcal{U}$ is a forgetful functor as above.

Definition 2.14 [place-sorted bigraphs [11]] An interface $\langle m, X\rangle$ is Θ-*place-sorted* if it is enriched by ascribing a sort θ to each place $i \in m$. A place sorted interface I is denoted by $\langle m, \overrightarrow{\theta}, X\rangle$ where $\overrightarrow{\theta} = (\theta_0, \ldots, \theta_{m-1})$ is a sequence of *sorts* where $sort(i) = \theta_i$ for $i \in m$. The underlying unsorted interface of I is denoted by $\mathcal{U}(I)$. $\acute{}\textsc{Big}(\mathcal{K}, \Theta)$ denotes the s-category of bigraphs in which the objects are place-sorted interfaces and each arrow $G : I \to J$ is a pure bigraph $G : \mathcal{U}(I) \to \mathcal{U}(J)$. The identities, composition, and tensor product are as in $\acute{}\textsc{Big}(\mathcal{K})$, but with sorted interfaces.

Definition 2.15 [place-sorting [11]] A *place-sorting* is a triple

$$\Sigma = (\mathcal{K}, \Theta, \Phi)$$

where Φ is a condition on Θ-sorted bigraphs over $\mathcal{K}$. The condition Φ must be satisfied by the identities and preserved by composition and tensor product.

A bigraph in $\acute{}\textsc{Big}(\mathcal{K}, \Theta)$ is Σ-*place-sorted* if it satisfies Φ. The Σ-sorted bigraphs form a sub-s-category of $\acute{}\textsc{Big}(\mathcal{K}, \Theta)$ denoted by $\acute{}\textsc{Big}(\Sigma)$. Further, if $\acute{}\mathcal{R}$ is a set of Σ-sorted reaction rules then $\acute{}\textsc{Big}(\Sigma, \acute{}\mathcal{R})$ is a Σ-*sorted* Brs.

Associated with a place-sorting is a forgetful functor $\mathcal{U} : `\textsc{Big}(\Sigma) \to `\textsc{Big}(\mathcal{K})$ which discards sorts. Such a functor $\mathcal{U}$ is called a *sorting* functor, is surjective on objects, and is faithful.

Kind bigraphs are an example of place-sorted bigraphs.

Proposition 2.16 *An s-category of kind or fitting bigraphs is place-sorted.*

The dynamics of a Brs is provided by a set of reaction rules. Figure 4 depicts a *parametric reaction rule* in a system where the controls are rooms (squares), enemies (hexagons), agents (triangles), and packages (circles). The rule states that an enemy in the same room as an agent may eliminate the agent. Reactions related to this rule fire when the missing bits, the parameters, are supplied. In a kind Brs, the redex and reactums of the reaction rules are kind bigraphs. The following informal definition ignores some technical details regarding parametric reaction rules. Those details are adressed in previous work [14].

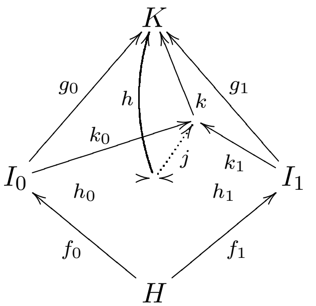
Definition 2.17 [kind reaction rule, kind Brs] A $\mathcal{K}$-sorted reaction rule is a pair R, R' of kind bigraphs over a signature $\mathcal{K}$. A kind bigraphical reactive system is a pair $\acute{}\textsc{Kbg}(\mathcal{K}, \acute{}\mathcal{R})$ consisting of an s-category of kind bigraphs $\acute{}\textsc{Kbg}(\mathcal{K})$ and a set $\acute{}\mathcal{R}$ of $\mathcal{K}$-sorted reaction rules.

Finally we recall the definition of relative pushout, the main universal construction in the theory. In the following, $\vec{f}$ will often be used to denote a pair f_0, f_1 of arrows in a s-category. If, for example, the two arrows share a domain H and have codomains I_0, I_1 we write $\vec{f} : H \to \vec{I}$.

Definition 2.18 [bound, consistent] Given four morphisms $\vec{f} : H \to \vec{I}$ and $\vec{g} : \vec{I} \to K$, if $g_0 f_0 = g_1 f_1$, then it is said that $\vec{g}$ is a *bound* for $\vec{f}$. If $\vec{f}$ has any bound, then it is said to be *consistent*.

Definition 2.19 [relative pushout (RPO)] Let $\vec{g} : \vec{I} \to K$ be a bound for $\vec{f} : H \to \vec{I}$. A *bound for* $\vec{f}$ *relative to* $\vec{g}$ is a triple $(\vec{h}, h)$ of arrows such that $\vec{h}$ is a bound for $\vec{f}$ and $h\ h_i = g_i (i = 0, 1)$. The triple may be called a *relative bound* when $\vec{g}$ is understood.

A *relative pushout (RPO) for* $\vec{f}$ *relative to* $\vec{g}$ is a relative bound $(\vec{h}, h)$ such that for any other relative bound $(\vec{k}, k)$ there is a unique arrow j for which $j\ h_i = k_i (i = 0, 1)$ and $k\ j = h$.



A relative pushout $(\vec{h}, h)$ for $\vec{f}$ relative to $\vec{g} : \vec{I} \to K$ is in fact a pushout in the slice category over K.

Definition 2.20 [idem pushout (IPO)] A pair $\vec{h} : \vec{I} \to J$ is an *idem pushout (IPO)* for the pair $\vec{f} : H \to \vec{I}$ if the triple $(\vec{h}, \mathrm{id}_J)$ is an RPO for $\vec{f}$ relative to $\vec{h}$.

The 'canonical' LTS we refer to in this document is called the *standard* transition system in the literature. IPOs provide the labels for this LTS. The idea is based on the definition of labelled transitions of Leifer and Milner [9] which was as follows: $a \xrightarrow{L} a'$ if there is a reaction rule (r, r') and a bigraph D for which (L, D) is an IPO for (a, r) and $a' = D \circ r'$. The current definition [7] is a refinement of this notion.

3 Properties of kind bigraphs

A kind bigraph is the same as a pure bigraph except that: i) a control may only contain controls that the signature specifies; and ii) as this property must hold under composition, the roots and certain leaves of the place graphs must also be sorted in a sensible way. We must now show that this generalisation does not break useful properties of bigraphs. We make much use of the forgetful functor $\mathcal{U}_{\mathcal{K}}$ to prove most of the properties we need. Its faithfulness is particularly useful.

3.1 Static properties

The notion of relative pushout is central to the theory of bigraphs. It is akin to a pushout in that it captures the notion of a minimal overlapping. As explained, IPOs play an important role in the operational semantics (LTS) of the theory. We now sketch the proofs that these constructions exist for kind and fitting bigraphs. The full proofs are presented in the technical report [14].

Proposition 3.1 *A kind (resp. fitting) RPO for $\vec{A} : H \to \vec{I}$ to $\vec{D} : \vec{I} \to L$ is constructed from a pure RPO for $\mathcal{U}_{\mathcal{K}}(\vec{A})$ to $\mathcal{U}_{\mathcal{K}}(\vec{D})$.*

Proof. We first construct a pure RPO $(\vec{B'} : \langle \vec{m}, X\rangle \to \langle n, Y\rangle, B' : \langle n, Y\rangle \to L)$ using Jensen and Milner's construction [7]. We use this to define a bound $(\vec{B}, B)$ for $\vec{A}$ to $\vec{D}$ by enriching the interface $\langle n, Y\rangle$. Intuitively, this interface should be least in some sense; in fact, the RPO will have an interface $\langle n, \vec{\theta}, Y\rangle$ such that $\vec{\theta}$ is the vector of least subsets of $\mathcal{K}$ which satisfies KR1 and KR2 in B_i.

We now have a kind (resp. fitting) bound $(\vec{B}, B)$ for $\vec{A}$ to $\vec{D}$. This relative bound is the kind (resp. fitting) RPO. □

The choice of interface in the proof above is the correct one but also fits with the intuition of colimits (*e.g.* pushouts) being minimal in some sense.

Proposition 3.2 *The forgetful functor $\mathcal{U}_{\mathcal{K}}$ associated with an s-category of kind or fitting bigraphs preserves RPOs* i.e. *if $(\vec{C}, C)$ is a binding RPO for $\vec{A}$ to $\vec{D}$ then $\mathcal{U}_{\mathcal{K}}(\vec{C}, C)$ is a pure RPO for $\mathcal{U}_{\mathcal{K}}(\vec{A})$ to $\mathcal{U}_{\mathcal{K}}(\vec{D})$.*

Proof. The proof is identical to the similar proof for binding bigraphs [7], noting that $\mathcal{U}_{\mathcal{K}}$ preserves isomorphisms. □

Definition 3.3 [consistency conditions for kind bigraphs] Let $\vec{A} : I \to \vec{J}$ be a pair of kind bigraphs with common inner face. We define three conditions for $\vec{A}$ to be consistent.

CP The consistency conditions for the underlying pure place graphs [8, Def. 7.9].

CL The consistency conditions for the link graphs [8, Def. 8.10].

KC If $A_i(w_2) \in V_i - V_2$ then $A_{\bar{\imath}}(w_2) \in n_{\bar{\imath}}$ and $\theta_{A_{\bar{\imath}}(w_2)} \subseteq kind(A_i(w_2))$, and if also $A_{\bar{\imath}}(w) = A_{\bar{\imath}}(w_2)$ then $w \in m \uplus V_2$ and $A_i(w) = A_i(w_2)$.

Theorem 3.4 (kind IPOs)

(i) *The consistency conditions* CP, CL *and* KC *are necessary for the existence of bounds in kind bigraphs.*

(ii) *If* $\vec{A}$ *has a kind IPO* $\vec{B}$ *with outer interface L then L is fitting for* $\vec{B}$.

(iii) *Let* $\vec{A}$ *satisfy the consistency conditions and* $\mathcal{U}_{\mathcal{K}}(\vec{A})$ *have a pure IPO* $\vec{B'}$. *Then* $\vec{A}$ *has a kind IPO* $\vec{B}$, *with* $\mathcal{U}_{\mathcal{K}}(\vec{B}) = \vec{B'}$.

(iv) *If* $\vec{A}$ *has a kind IPO* $\vec{B}$, *then* $\mathcal{U}_{\mathcal{K}}(\vec{A})$ *has a pure IPO* $\mathcal{U}_{\mathcal{K}}(\vec{B})$.

As a corollary of Theorems 3.4.iii and 3.4.iv above, when a pair $\vec{A}$ of kind bigraphs is consistent, there is a precise correspondence between its kind IPOs and the pure IPOs of $\mathcal{U}_{\mathcal{K}}(\vec{A})$. This is a useful result; the labelled transitions of a Brs arise from the IPOs and the IPOs of pure Brss have a characterisation [7]. This allows us to enumerate over the possible labels of an agent (a bigraph without any sites).

Much of the remainder of the static theory of pure bigraphs by Jensen and Milner [7] has also been generalised to the setting of kind bigraphs [14].

3.2 Dynamics

Leifer and Milner [10] identified the following two properties and proved that if the forgetful functor of a link-sorting satisfies them then bisimilarity results from the pure theory can be transferred to the sorted theory. Milner [11] proved a similar proposition for place-sortings.

Definition 3.5 [creating RPOs, reflecting pushouts [11]] Let $\mathcal{F}$ be any functor on an s-category $\acute{\mathbf{A}}$. Then $\mathcal{F}$ *creates RPOs* if, whenever $\vec{D}$ bounds $\vec{A}$ in $\acute{\mathbf{A}}$, then any RPO for $\mathcal{F}(\vec{A})$ relative to $\mathcal{F}(\vec{D})$ has a unique $\mathcal{F}$-preimage that is an RPO for $\vec{A}$ relative to $\vec{D}$.

$\mathcal{F}$ *reflects pushouts* if, whenever $\vec{D}$ bounds $\vec{A}$ in $\acute{\mathbf{A}}$ and $\mathcal{F}(\vec{D})$ is a pushout for $\mathcal{F}(\vec{A})$, then $\vec{D}$ is a pushout for $\vec{A}$.

We have the following positive results.

Proposition 3.6 *The forgetful functor* $\mathcal{U}_{\mathcal{K}}$ *of any kind or fitting Brss creates RPOs. The forgetful functor* $\mathcal{U}_{\mathcal{K}}$ *of any fitting Brss reflects pushouts.*

Corollary 3.7 *In a kind or fitting Brs, bisimulation on the standard transition system is a congruence.*

With respect to what we have introduced in this paper, *simple prime affine* means that for each redex of each reaction rule; the outer interface has width 1, the parent of any site is a node, and no two sites are siblings.

Corollary 3.8 *In a fitting Brs with simple prime affine reaction rules, we can canonically reduce*[4] *the standard transition system without affecting bisimilarity.*

[4] A technical definition is beyond the scope of this paper.

3.3 Homomorphic sortings as kind Brss

A homomorphic sorting is a place sorting with the property that the children of a root or node all have the same sort and, further, a root has the same sort as all of its children. Milner introduced homomorphic sortings in his encoding of finite CCS [11].

Definition 3.9 [homomorphic sorting] In a *homomorphic* sorting $\Sigma = (\mathcal{K}, \Theta, \Phi)$ the condition Φ assigns a sort $\theta \in \Theta$ to each control in $\mathcal{K}$. It also defines a parent map $prnt : \theta \to \Theta$ over sorts. Then a bigraph is admissible iff, for each site or node w,

– if $prnt(w)$ is a node then the sort assigned to its control is $prnt(\theta)$;

– if $prnt(w)$ is a root then its sort is θ.

Proposition 3.10 (homomorphic sorting is well behaved [11]) *Every homomorphic sorting creates RPOs and reflects pushouts.*

A homomorphic sorting can be described as a kind signature as follows. Let n be the cardinality of Θ. For each sort $\theta \in \Theta$, partition $\mathcal{K}$ into n disjoint subsets $\mathcal{K}_\theta$ of $\mathcal{K}$ such that if control K has sort θ then $K \in \mathcal{K}_\theta$. To take care of the parent map $prnt : \Theta \to \Theta$ (not to be confused with the $prnt$ map of a bigraph), if $prnt(\theta) = \theta'$ then for each $K \in \mathcal{K}_{\theta'}$, let

$$kind(K) = \emptyset \text{ if } K \text{ is atomic}$$
$$kind(K) = \mathcal{K}_\theta \text{ otherwise.}$$

We have now satisfied most of the conditions of a homomorphic sorting. The final condition is that roots are sorted. In general, places of kind interfaces can be defined to contain any subset of $\mathcal{P}(\mathcal{K})$. We wish to restrict this so that if $r \in m$, for some interface of width m, then $kind(r) = \mathcal{K}_\theta$ for some sort θ. We therefore restrict ourselves to the full sub-s-category of kind bigraphs over $\mathcal{K}$ whose interfaces satisfy this property. We will call this sub-s-category of $\text{KBG}(\mathcal{K})$ the *homomorphic s-category* and denote it by $\text{KBG}(\Sigma)$. $\text{KBG}(\Sigma)$ is defined when $\mathcal{K}$ is a kind signature derived as above from a homomorphic sorting Σ.

In general, the forgetful functor from kind bigraphs to pure bigraphs does not reflect pushouts. However, the functor $\mathcal{U}_\mathcal{K} : \text{KBG}(\Sigma) \to \text{BIG}(\mathcal{K})$ from a homomorphic s-category does. It also creates RPOs and so the bisimilarity results of Corollaries 3.7 and 3.8 follow.

4 Dynamics and expressivity

The dynamics of a Brs is provided by a set of reaction rules. As kind bigraphs have extra structure to pure bigraphs, the reaction rules are more expressive. Some of this expressivity is expected but there are some interesting consequences.

4.1 Subject reduction

In a kind Brs, the sorting is preserved by reaction (subject reduction). This is largely unsurprising and is a result of two facts. First, the redex and reactum of the rules are sorted. Secondly, the pure theory requires that the outer interface of the redex and reactum must be the same (interface preservation). The sorted theory requires that the reactum may permute the sites of the redex but that copies of a redex site must retain their sorting. Subject reduction then falls out from the definitions of composition and the dynamic theory.

4.2 An expression of absence

A kind reaction rule has more structure than a pure rule in one important respect – the interfaces are sorted. For the outer interface, this does not mean much; any interface satisfying the kind rules for both redex and reactum is adequate and we typically let a single reaction rule represent a rule schema in this regard. However, we can make some interesting choices when sorting the inner interface.

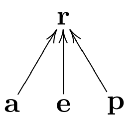
For example, Figure 1 depicts a parametric kind reaction rule in a model of a smart building where the sort of site 0 is $\{\mathbf{PC}\}$. Therefore, any parameter to this rule which will fill this hole can not contain a person *i.e.* we are able to specify the *absence* of nodes of certain controls in parametric kind reaction rules. Therefore, the rule in Figure 1 models the behaviour 'when the last person leaves the room, the light is switched off.'

The expressivity lies in the fact that we can guarantee the absence of certain controls in the set of possible parameters of a reaction rule. There does not seem to be any way to express these types of rules in the pure theory. We now demonstrate how this extra expressivity allows us to add some level of flow-control to the dynamics of a system.

4.3 Prioritised reaction firing

A side-effect of this extra structure can lead to what we call prioritised reaction firing. It is not a feature of all kind Brss but it may be possible to take advantage of it in some cases.

For example, take the kind Brs over the signature



where *rooms* may contain *agents*, *enemies*, and *packages*. Figures 2–5 depict the parametric kind reaction rules for this system where the sites have restricted sorts: in Figure 2, the site 1 can not contain enemies or packages; in Figure 3, the site 1 can not contain an agent; and in Figure 4, the single site can not contain a package.

These rules may be restated conditionally *e.g.* Figure 4 models the behaviour where an enemy in the same room as an agent may eliminate the agent *so long as* there are no packages in the area.

The collection of these rules allows us to describe the behaviour of an agent as:

```
if package in room, collect package
else if enemy in room, die
else move.
```

The rule-set, at least from the perspective of agent nodes, has some flow control built in. This is what we mean by 'prioritised reaction firing.' More small examples are given in the technical report [14].

5 Summary

We have presented a sorting of pure bigraphs borne from the desire to model containment in 'real-world' hierarchies. While our examples have concentrated on these hierarchical sortings, the type of $kind : \mathcal{K} \to \mathcal{P}(\mathcal{K})$ is quite free and we only force constraints on parent-child relationships. The sorting is also expressive enough to encompass the homomorphic sortings used to encode finite CCS.

A standard labelled transition system based on minimal transitions (IPOs) can be derived for kind Brss. The desirable properties of pure bigraphs relating to bisimilarity carry over to the sorted theories although the results seem stronger for fitting bigraphs.

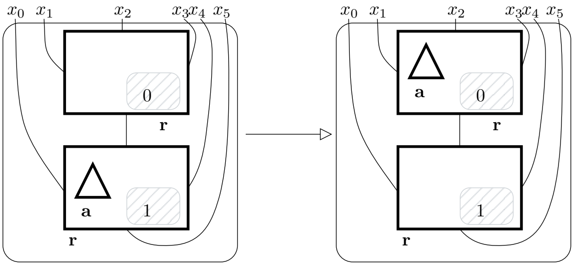
Finally, we have demonstrated an extra expressiveness in kind parametric reaction rules and shown how some systems can be designed with a level of flow control in the rules.

5.1 Future directions

We first introduce an informal scenario inspired by Cardelli and Gordon [4]. We have a kind Brs modelling an office building with rooms, key holders, thieves, and locked and unlocked doors. Reaction rules encode the following behaviours: key holders outside a locked room for which they have a key may enter the room, leaving it unlocked; key holders may exit an unlocked room (leaving it unlocked); and thieves may enter an unlocked room *so long as* there are no key holders inside. This last rule takes advantage of the expressiveness of Section 4.2.

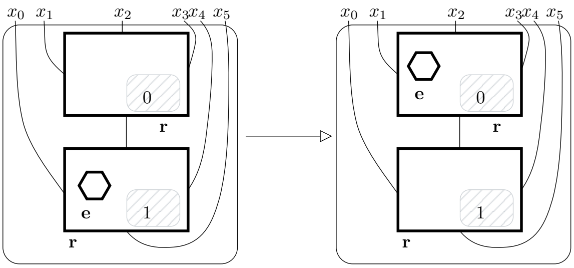
We wish to define a logic where we could prove that beginning from a state where all the doors were locked, all the keyholders were outside the rooms, and a thief was present, at some time and place, a thief will gain access to an unlocked room. This would prove, for example, that the Brs does not satisfy some security specification. This logic would be a modal logic.

Spatial logics [4,3], modal in both time and space, have recently been investigated in the bigraphical setting [5]. The spatial modality is very interesting for reasoning about distributed systems. It allows logical statements like "at some place in process P, formula $\mathcal{A}$ holds." Other modal operators defined in the cited works include "everywhere", "sometime", and "somewhere." Combined with other logical operators (classical, tensor, quantification), the logics are very expressive, and can



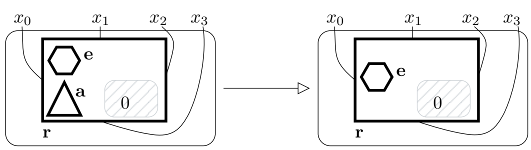
$$/l(\mathbf{r}_l\Box|\mathbf{r}_l(\mathbf{a}|_{|\neg\mathbf{e}\neg\mathbf{p}|})) \longrightarrow\!\!\triangleright\, /l(\mathbf{r}_l(\mathbf{a}|\Box)|\mathbf{r}_{l\,|\neg\mathbf{e}\neg\mathbf{p}|})$$

Fig. 2. An agent moves between two rooms



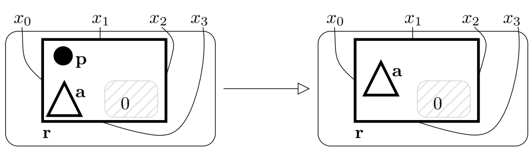
$$/l(\mathbf{r}_l\Box|\mathbf{r}_l(\mathbf{e}|_{|\neg\mathbf{a}|})) \quad \triangleright\, /l(\mathbf{r}_l(\mathbf{e}|\Box)|\mathbf{r}_{l\,|\neg\mathbf{a}|})$$

Fig. 3. An enemy moves between two rooms



$$\mathbf{r}(\mathbf{a}|\mathbf{e}|_{|\neg\mathbf{p}|}) \quad \triangleright\, \mathbf{r}(\mathbf{e}|_{|\neg\mathbf{p}|})$$

Fig. 4. An enemy eliminates an agent



$$\mathbf{r}(\mathbf{a}|\mathbf{p}|\Box) \quad \triangleright\, \mathbf{r}(\mathbf{a}|\Box)$$

Fig. 5. An agent collects a package

handle fresh name quantification, name hiding, and recursion [3].

Kind bigraphs constrain the spatial structure of bigraphs. We feel that a modal logic for kind bigraphs may benefit from this. Reaction rules such as those in Section 4.2 which express the absence of controls come with 'free' logical formulae *e.g.* any parameter to this redex does not contain any nodes of control K.

5.2 Related work

Kind bigraphs are an example of a well-behaved place-sorting. A sorting on the link graph structure was used by Leifer and Milner [10] in their encoding of condition-event Petri nets. In other work [14], we introduced a sorting with similarities to Leifer and Milner's directed linear link-sorting. Bundgaard and Sassone [2] have used a novel kind of link sorting to represent capability types of the typed polyadic π-calculus, and so provide a foundation for presenting related type systems in Brss.

6 Acknowledgements

The beginnings of this work, based on a remark of Jensen and Milner [8], arose during a visit at the IT University (ITU), Copenhagen. The visit was made possible by the support of the author's supervisor, Mícheál Mac an Airchinnigh, and Lars Birkedal, ITU. Discussions with the Bigraphical Programming Languages (BPL) group were very useful and particular thanks are due to Robin Milner and Søren Debois for their comments and suggestions on this work.

References

[1] Lars Birkedal, Mikkel Bundgaard, Troels Christoffer Damgaard, Søren Debois, Ebbe Elsborg, Arne John Glenstrup, Thomas Troels Hildebrandt, Robin Milner, and Henning Niss. Bigraphical programming languages for pervasive computing. In Thomas Strang, Vinny Cahill, and Aaron Quigley, editors, *Proceedings of Pervasive 2006 International Workshop on Combining Theory and Systems Building in Pervasive Computing*, pages 653–658, May 2006.

[2] Mikkel Bundgaard and Vladimiro Sassone. Typed polyadic pi-calculus in bigraphs. In *Proceedings of the 8th International Symposium on Principles and Practice of Declarative Programming (PPDP'06)*, 2006.

[3] Luís Caires and Luca Cardelli. A spatial logic for concurrency (part I). *Information and Computation*, 186(2):194–235, 2003.

[4] Luca Cardelli and Andrew D. Gordon. Anytime, anywhere: modal logics for mobile ambients. In *POPL '00: Proceedings of the 27th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 365–377, New York, NY, USA, 2000. ACM Press.

[5] Giovanni Conforti, Damiano Macedonio, and Vladimiro Sassone. Spatial logics for bigraphs. In *Proceedings of International Colloquium on Automata, Languages and Programming*, 2005.

[6] Ole Høgh Jensen. *Mobile Processes in Bigraphs.* PhD thesis, Department of Computer Science, Aalborg University, 2006.

[7] Ole Høgh Jensen and Robin Milner. Bigraphs and transitions. In *POPL '03: Proceedings of the 30th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 38–49, New York, NY, USA, 2003. ACM Press.

[8] Ole Høgh Jensen and Robin Milner. Bigraphs and mobile processes (revised). Technical Report UCAM-CL-TR-580, Computer Laboratory, University of Cambridge, February 2004.

[9] James J. Leifer and Robin Milner. Deriving bisimulation congruences for reactive systems. In Catuscia Palamidessi, editor, *CONCUR*, volume 1877 of *Lecture Notes in Computer Science*, pages 243–258. Springer, 2000.

[10] James J. Leifer and Robin Milner. Transition systems, link graphs and petri nets. Technical Report UCAM-CL-TR-598, Computer Laboratory, University of Cambridge, August 2004.

[11] Robin Milner. Pure bigraphs. Technical Report UCAM-CL-TR-614, Computer Laboratory, University of Cambridge, 2005.

[12] Robin Milner. Local bigraphs and confluence: two conjectures. In Roberto Amadio and Iain Phillips, editors, *Proceedings of the 13th International Workshop on Expressiveness in Concurrency (EXPRESS '06)*. Electronic Notes in Theoretical Computer Science, 2006.

[13] Robin Milner. *Pure bigraphs: a tutorial*. TBA, 2007.

[14] Shane O'Conchúir. Kind bigraphs – Static theory. Technical Report TCD-CS-2005-36, Trinity College Dublin, May 2005.

A Categorical definitions

Definition A.1 [tensor product, monoidal precategory] A *(strict, symmetric) monoidal* precategory has a partial *tensor product* $\otimes$ both on objects and on arrows. It has a unit object ϵ, called the *origin*, such that $I \otimes \epsilon = \epsilon \otimes I = I$ for all I. Given $I \otimes J$ and $J \otimes I$ it also has a *symmetry* isomorphism $\gamma_{I,J} : I \otimes J \to J \otimes I$. The tensor and symmetries satisfy the following equations when both sides exist:

(1) $f \otimes (g \otimes h) = (f \otimes g) \otimes h$ and $\mathrm{id}_\epsilon \otimes f = f$

(2) $(f_1 \otimes g_1)(f_0 \otimes g_0) = (f_1 f_0) \otimes (g_1 g_0)$

(3) $\gamma_{I,K} \circ (f \otimes g) = (g \otimes f) \circ \gamma_{H,J}, f : H \to I, g : J \to K.$

(4) $\gamma_{I,\epsilon} = \mathrm{id}_I$

(5) $\gamma_{J,I} \circ \gamma_{I,J} = \mathrm{id}_{I \otimes J}$

Definition A.2 [s-category] An *s-category* $\acute{\mathbf{C}}$ is a strict symmetric monoidal precategory which has:

- for each arrow f, a finite set $|f|$ called its *support*, such that $|\mathrm{id}_I| = \emptyset$. For $f : I \to J$ and $g : J \to K$ the composition gf is defined iff $|g| \cap |f| = \emptyset$ and $\mathrm{dom}(g) = \mathrm{cod}(f)$; then $|gf| = |g| \uplus |f|$. Similarly, for $f : H \to I$ and $g : J \to K$ with $H \otimes J$ and $I \otimes K$ defined, the tensor product $f \otimes g$ is defined iff $|f| \cap |g| = \emptyset$; then $|f \otimes g| = |f| \uplus |g|$.
- for any arrow $f : I \to J$ and any injective map ρ whose domain includes $|f|$, an arrow $\rho \bullet f : I \to J$ called a *support translation* of f such that:

(1) $\rho \bullet \mathrm{id}_I = \mathrm{id}_I$

(2) $\rho \bullet (gf) = (\rho \bullet g)(\rho \bullet f)$

(3) $\rho \bullet (f \otimes g) = \rho \bullet f \otimes \rho \bullet g$

(4) $\mathrm{id}_{|f|} \bullet f = f$

(5) $(\rho_1 \circ \rho_0) \bullet f = \rho_1 \bullet (\rho_0 \bullet f)$

(6) $\rho \bullet f = (\rho \restriction |f|) \bullet f$

(7) $|\rho \bullet f| = \rho(|f|)$.

Each equation is required to hold only when both sides are defined.

Definition A.3 [sub-s-category] A functor $\mathcal{F} : \acute{\mathbf{C}} \to \acute{\mathbf{D}}$ defines a *sub-s-category* when $\mathbf{D}$ is an s-category and $\mathcal{F}$ is injective on objects and homsets. $\mathcal{F}$ defines a *full* sub-s-category iff $\mathcal{F}$ is bijective on homsets.